

Automatic Key Detection

Computer Music Seminar

Leon Wittwer

June 28, 2017

Table of Contents

Introduction

Theory of Tonality and Key

Key Detection

Symbolic key detection

Audio Key Detection

Approach in Thesis

Music Playing

Conclusion

Introduction

Motivation

- Important characteristic of musical pieces
- Large digital collections of music, not feasible to annotate key by hand
- Music Perception
- Improves chord recognition systems
- Automated mixing

Challenges

- Key recognition is even challenging for humans
- Tuning Variations
- Low Frequency Resolution
- Effect of Partial
- Modulations (Change of the Key within a piece)

Theory of Tonality and Key

Definition of Key:

Key is "the pitch relationships that establish a single pitch-class as a tonal center or tonic (or key note), with respect to which the remaining pitches have subordinate functions" [Oxford Dictionary of Music]

- two modes (major, minor)
- tonic (one of twelve pitch-classes)

Common Errors (Explained)

- Perfect 5th Errors
 - a tonic that is detected seven semitones away from the correct tonic. Only one pitch in the class is not the same (but near).



Figure 1: C Dur Scale

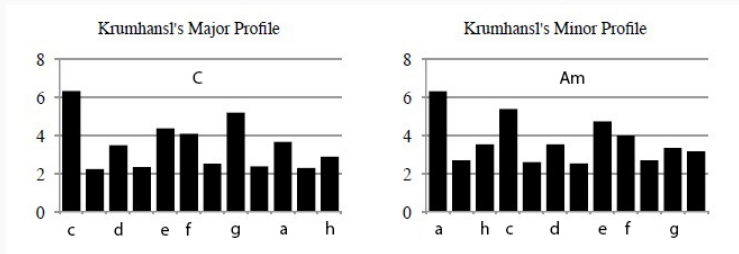


Figure 2: G Dur Scale

- Relative Major / Minor Errors
- Parallel Major / Minor Errors

Common Errors (Explained)

- Perfect 5th Errors
- Relative Major / Minor Errors
 - The pitch class is the same, only the number a note appears and the relations do change.



- Parallel Major / Minor Errors

Common Errors (Explained)

- Perfect 5th Errors
- Relative Major / Minor Errors
- Parallel Major / Minor Errors
 - Same tonic: A vs. Am



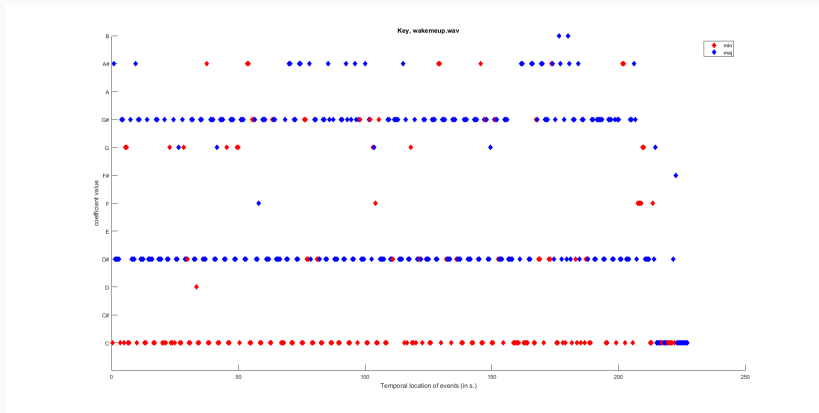
Figure 3: A Dur Scale



Figure 4: A Minor Scale

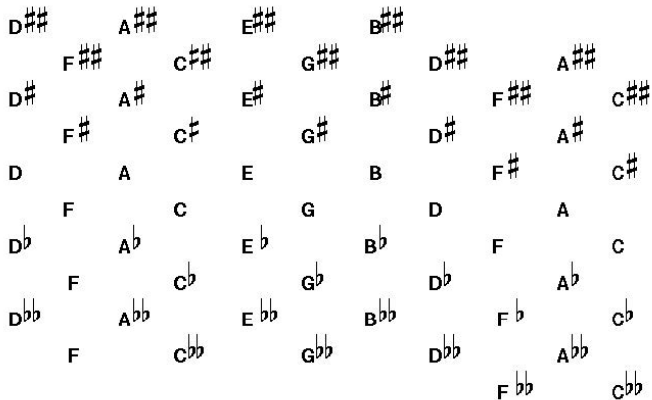
Common Errors (Example)

Music Playing: Wake me up (Johnny May)

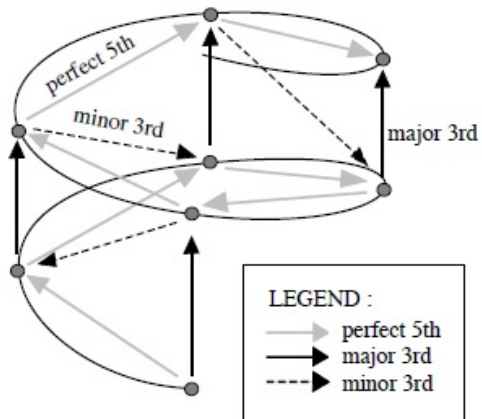


Correct key: Dis Dur, so Gis Dur is the perfect fifth error and C minor is the relative minor error.

Harmonic Network [1]



Spiral Array Model



Key Detection

Key detection can be divided in **symbolic** and **audio** key detection.

- symbolic key detection
 - uses symbolic description of music, like scores and MIDI files
 - emerged earlier (1971 vs. 1991)
- audio key detection
 - uses audio files
 - added difficulty of analyzing audio
 - less documented research

Symbolic key detection

Symbolic Key Detection

The first approach to symbolic key detection was done by Longuet-Higgins and Steedman in 1971

- shape matching algorithm on the Harmonic Network

C major

B	F \sharp	C \sharp	G \sharp	D \sharp	A \sharp	E \sharp
G	D	A	E	B	F \sharp	C \sharp
E \flat	B \flat	F	C	G	D	A
C \flat	G \flat	D \flat	A \flat	E \flat	B \flat	F

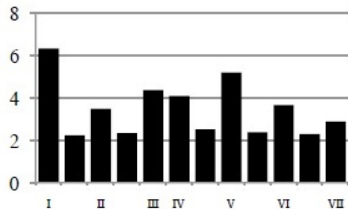
A minor

B	F \sharp	C \sharp	G \sharp	D \sharp	A \sharp	E \sharp
G	D	A	E	B	F \sharp	C \sharp
E \flat	B \flat	F	C	G	D	A
C \flat	G \flat	D \flat	A \flat	E \flat	B \flat	F

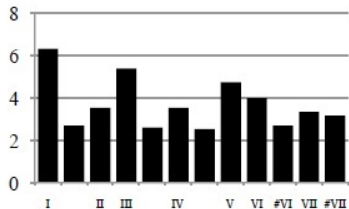
Krumhansl's major and minor key profiles

- Next big step: key profiles derived by experiments
- Krumhansl and Schmuckler in 1990
- The key profiles represent the ideal distribution of pitch-classes within a key

Krumhansl's Major Profile



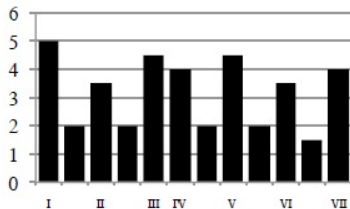
Krumhansl's Minor Profile



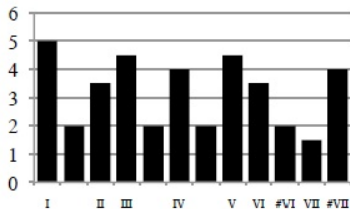
Temperley's major and minor key profiles

- Temperley in 1999
- proposed modifications to the key profiles for better distinguishing

Temperley's Major Profile



Temperley's Minor Profile



Audio Key Detection

It is possible to group audio key detection systems in this four categories:

- pattern matching and score transcription methods
- template-based models
- geometric models
- models based on chord progressions or HMMs

- Leman in 1991
 - one of the first models for audio key detection
 - pattern matching based approach
 - extract tone centers and compare with predetermined templates
- Izmirli and Bilgen in 1994
 - uses partial score transcription and pattern matching

- template based method
 1. extract pitch-class distributions
 2. compare the extracted distributions with pitch-class templates
- create three different distributions from the audio using different temporal weighting functions
- uses Krumhansl's key profile as template

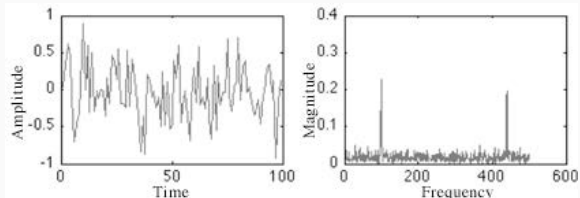
Approach from Lee and Slaney (2007)

- HMM-based system
- performs chord recognition and key detection simultaneously
- uses tonal centroid vector
- 24 separate HMM's with 24 states each was used
 - each HMM was trained for one of the 24 possible keys
 - each state should represent a single type of chord (major / minor)

Approach in Thesis

Feature Extraction

- Frequency analysis
 - Use Fast Fourier Transform to transform the audio signal from the time domain to the frequency domain



- Pitch class extraction
 - Basic Mapping
 - Peak detection extension
 - Spectral flatness measure
 - Low frequency clarification
- Pitch class aggregation

- Use a mapping matrix $M_{i,j}$ to create pitch class distribution vector p_i from the FFT result x_j , where $j = 0, \dots, N$ with length of analyzed window N :

$$p_i = \sum_{j=0}^N M_{i,j} \cdot x_j$$

- The mapping matrix $M_{i,j}$ is created using a gaussian distribution function. **page 44 in [1], not readable.**

$$M_{i,j} = e^{-\frac{1}{2}(2D_{i,j})^2}$$

Basic Mapping

- The $12 \times N$ matrix D contains the projected values of $n(f)$ for each pitch class from -6 to +6.

For : $i = 0, \dots, 11$

$$D_{i,j} = ((n(f_j) - i + 6) \bmod 12) - 6$$

- $n(f_i)$ is used to map the frequency to a note.

$$n(F_i) = 12 \log_2 \left(\frac{f_i}{f_0} \right)^2$$

Feature Extraction

- Pitch class extraction
 - Basic Mapping
 - Peak detection extension
 - Only peaks are counted. Peaks are FFT values that are greater than the average value in the neighbourhood.
 - Spectral flatness measure
 - The spectral flatness measure is based on arithmetic and geometric means and is employed to also ensure that only peaks and no noise is taken into account.
 - Low frequency clarification
 - Due to low resolution in the low frequencies peaks are eliminated if a neighbouring peak has a greater value. So the effect of spectral leakage is not considered to be a single note.
- Pitch class aggregation
 - To counteract accumulating errors the mean has to be reset to zero.

I would like to show recognition results from the GiantSteps data set: <http://www.cp.jku.at/datasets/giantsteps/>
Because the evaluation in the paper is more focused on the different parts of their own system, which I do not explain. Furthermore the GiantSteps Dataset is Electronic Dance Music and the systems that are evaluated are some recently updated DJ software, so this is more up to date and the results (70% best recognition) does show that there is much room for improvement.

Music Playing

Planned

- Live comparison of at least two different approaches:
 - one piece that both gets right, one piece no one gets right
- Maybe letting the audience guess whats right predicted, whats wrong predicted
- If there is a difference in recognition of self recorded and midi generated pieces this is a nice example I think, so i will show it

An already existing implementation is the MIRToolBox for MATLAB: <https://www.jyu.fi/hytk/fi/laitokset/mutku/en/research/materials/mirtoolbox>, which is very nice because it provides different visualization tools. KeyFinder: <http://www.ibrahimshaath.co.uk/keyfinder/> to compare the approaches.

Conclusion

- many methods are proposed to do key recognition
- but nevertheless is it hard to detect the correct key of a musical piece
- so no completely reliable approach to detect keys is known



Spencer Campbell.

Automatic key detection of musical excerpts from audio.

Master's thesis, 2010.

Some prepared slides for questions