AUTOMATIC CHORD DETECTION
INCORPORATING BEAT AND KEY DETECTION

Veronika Zenz, Andreas Rauber, Member, IEEE

Vienna University of Technology
Department of Software Technology and Interactive Systems
Favoritenstrasse 9-11/188, A-1040 Vienna, Austria
www.ifs.tuwien.ac.at/mir

ABSTRACT

We introduce an algorithm that analyzes audio signals to extract chord-sequence information. The main goal of our approach lies in incorporating music theoretical knowledge without restricting the input data to a narrow range of musical styles. At the basis of our approach lies pitch detection using enhanced autocorrelation, supported by key detection and beat tracking. The Chords themselves are identified by comparing generated and reference Pitch Class Profiles. A smoothing algorithm is applied to the chordsequence which optimizes the number of chord changes and thus takes into consideration the comparatively stable nature of chords. In this paper we present an evaluation performed on a large set of 35 pieces of diverse music showing an average performance of 65% accuracy.

Index Terms— Acoustic signal analysis, Audio systems, Music

1. INTRODUCTION

Automatic chord detection is part of the large research field of computer audition (CA) which deals with all kinds of information extraction from audio signals. Chord detection extracts the harmonies that occur over the time of a piece of music.

Figure 1 depicts an exemplary result of chord detection visualized with Audacity. Chord detection is a special form of lossy music transcriptions, that captures only harmonic properties of the audio signal. It is particularly interesting as chords are comparatively simple and stable structures, and at the same time completely describe a piece of music in terms of occurring harmonies. The great interest of musicians in chord sequence is perhaps best demonstrated by pointing out the large number of websites, newsgroup and forum messages on this topic. Newsgroups like rec.music.makers.guitar.tablature, or alt-.guitar.tab offer a platform to request and publish chord sequences and tablatures together with the lyrics of songs.

Music theorists apply harmonic analysis to gain knowledge about the compositional structure and the artistic intent of an opus. An example for such an analysis including functional signs below and absolute chords above the stave is shown in Fig. 2.

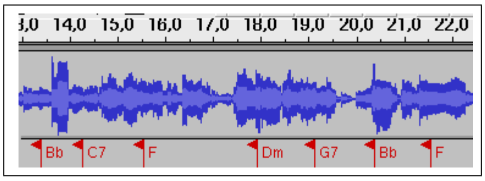

Figure 1. Detected Chords Output

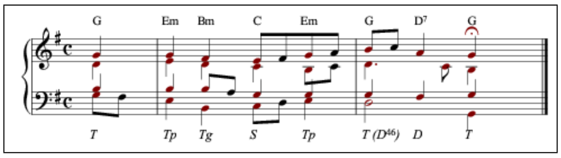

Figure 2. Harmonic Analysis

Besides transcription and analysis as an end in themselves, a new application field has evolved in the last years: Music Information Retrieval (MIR). Formats such as mp3 and decreasing storage costs lead to the rise of large private and commercial digital music collections. In order to search these archives for audio with special properties, each music file has to be annotated with this information. Such properties are usually artist, title and genre, but could as well be mood, melody, harmonies, lyrics and so on. Manual information extraction and annotation is rather time consuming, thus implying the need to design algorithms that compute these features automatically. The chordsequence of a song does not only describe its harmonic properties but can also be used to draw conclusions on its genre and emotions that are evoked at the listener. Thus, having a music database annotated with chord information, users could search for specific chordsequences, music with complex or simple chord structures, slow or fast chord progressions, rather sad (minor) or lively (major) music and so on.

2. RELATED WORK

A number of approaches on chord detection have been reported in literature. In [4] Hidden Markov Models are used to represent chord sequences. Transition and output probabilities are computed using an expectation maximization algorithm on hand-labelled chord sequences. The authors trained the algorithm with 18 Beatles songs and evaluated it using 2 other songs from the Beatles with a result of 23% chord recognition accuracy.

Multi-timbre Chord Classification Using Wavelet Transform and Self-organized Map Neural Networks are used in [5]. This is one of the few approaches that does not use Pitch Class Profiles but evaluates the frequency spectrum directly. The results of a wavelet transform are directly sent to a neural-network chord-classification unit without note identification. An accuracy rate of 100% on a testset of 8 measures of a Beethoven Symphony is reported.

The emphasis of the work reported in [7] is on the mutual dependency of chord-boundary detection and chord symbol identification. For their solution they do not only use frequency based features but also beat detection and a high-level database of common chord sequences. The heart of this algorithm is a hypothesis-search algorithm that evaluates tuples of chord symbols and chord boundaries. The system was tested on excerpts of seven Pop-songs taken from the RWC Music Database for which an average accuracy of 77% has been achieved.

3. AUTOMATIC CHORD DETECTION

3.1. Algorithm Overview

Figure 3 depicts the flow chart diagram of our chord detection algorithm. The algorithm takes sampled audio as input and outputs a sequence of time-chord pairs. It has a modular design, the four main modules being beat tracking, key detection, chord detection and chord sequence smoothing. Beat tracking is used to split the audio data into blocks of sizes that correspond to the computed beat structure. In the chord detection module each of the blocks obtained by beat tracking is passed to an enhanced autocorrelation algorithm which calculates the frequency spectrum and returns a sequence of frequency-intensity pairs. These are then used to compute the intensity of each pitch class, the so called Pitch Class Profile (PCP). The key detection module extracts the key of the input song which is then used to refine the set of possible chords. The calculated PCP's are then compared to this filtered set of reference chord PCP's. For each timespan the most probable chords are saved into a chord sequence matrix. A smoothing algorithm is applied to this matrix which rates each chord according to the number of chord changes around it. Finally, for each timespan the chord with the highest score is taken.

3.2. Tempo and Beat Tracking

Instead of slicing the song into equally-sized blocks of data, a tempo and beat tracking algorithm is used to obtain the beat of the song. As we know that chord changes usually occur at beat time, this is an efficient method to lengthen the span of the audio-blocks without risking windows that span over two or multiple chords. For this purpose the BeatRoot[1] program is utilized, which ranked best in the MIREX 2006 Audio Beat Tracking Evaluation[2].

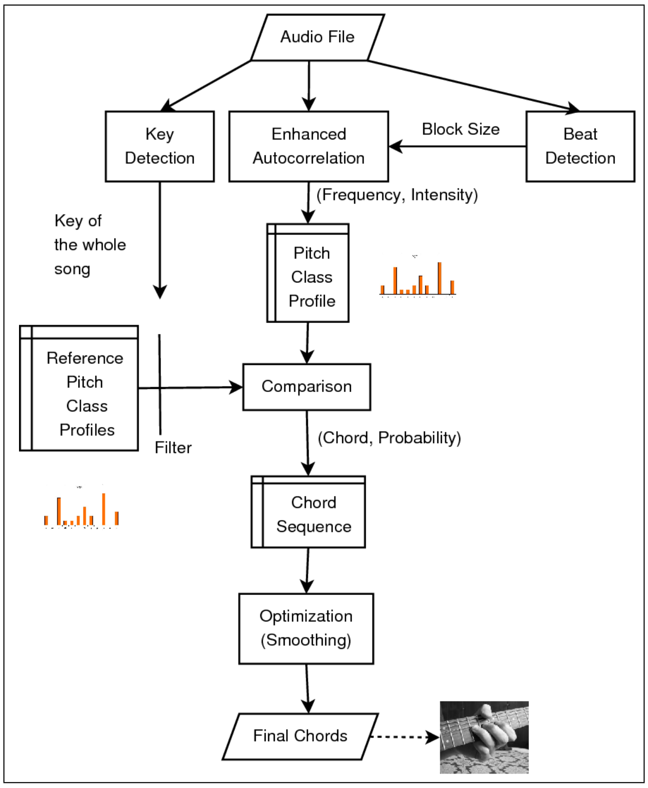

Figure 3. Flow Chart of our Chord Detection Algorithm

The tracking algorithm consists of three subsystems: Onset detection, tempo induction and beat tracking. Onset detection of musical events (notes, percussive beats) is performed by finding peaks in the spectral flux. Onset times are passed to a tempo induction algorithm that calculates initial hypotheses for the tempo by computing the inter-onset intervals. These hypotheses are then used to initialize multiple beat tracking agents. From the beginning of the music to the end each agent then predicts the next beat. Onsets which correspond to an inner window of predicted beat times are taken as actual beats and used to adapt the agent's tempo and phase. An evaluation function is used to rate each agent and the beat track of the agent with the highest score is finally taken as the solution to the beat tracking problem [1],[2].

3.3. PCP Generation

A chord is an accord of specific notes that lasts for a certain time. The detection of the pitches, that is at the first instance the detection of the frequencies, that occur over the time is therefore fundamental for our chord detection algorithm. For multipitch detection we have chosen the Enhanced Autocorrelation (EAC) Algorithm described by Tolonen et al. in [6], which offers a good trade-off between complexity and quality. The EAC function is ap-

[1] http://www.elec.qmul.ac.uk/people/simond/beatroot/

[2] http://www.music-ir.org/mirexwiki/index.php/MIREX_2006

plied to non-overlapping successive blocks of the input data. The size of each block is calculated using the Beat Detection Algorithm as described in the previous section. Each input block is split into equally sized overlapping windows. In our tests a Hamming window of 46.4ms (that is 1024samples for a sampling rate of 22050Hz) and a window-overlap of 50%, corresponding to a hop size of 23.2 ms, gave the best results.

From the enhanced autocorrelation the PCP are derived, introducing two layers of abstraction: The spectral energy is summarized into frequency bands of one semitone width. Additionally, all pitches are folded into a single octave, the pitch class.

The generated PCPs are compared to a set of reference chord PCPs. This is done by calculating the linear distance between the two normalized vectors. The smaller the distance, the higher the resemblance of the two PCPs. The reciprocal of the distance is then stored as a reference value for the probability that the chord at this timespan matches. Finally, for each timespan k chords with the highest probabilities are stored in a chordsequence-matrix.

3.4. Reference PCPs

This work distinguishes three chord types: Major, Minor and Diminished. Reference PCPs of the corresponding seventh chord are also used, though in evaluation stage the confusions between the standard and the seventh version of the chords are not counted as an error. The reference key PCPs have been determined empirically, A seperate dataset of one minute excerpts of 5 popular songs was used to obtain reference PCPs for the different chordtypes. On this dataset PCPs were computed and, knowing the ground truth, shifted to the same root. The average of all PCPs for one chordtype was then normalized and adapted manually to assure that the value of their root pitches did not vary. We used this adaptation rule to remove inherent biases towards one or another chordtype in the quite frequent case that only one note is present in the PCP to be analyzed: if, for example, only 'c' is present in a PCP, the reference PCPs of C-Major and C-Minor chords then have the same distance and are considere equiprobable by our system. For each chord type one single reference PCP is stored. The 12 chords of this chord type can then be obtained by shifting the PCP by one step for each semitone.

3.5. Key Detection

For key detection we largely reuse the algorithm for chord detection: The PCPs of the first and last 15 seconds of the input data are used to calculate a mean PCP which is then compared to the reference PCPs. The key whose reference PCP has minimal distance with the calculated mean PCP is selected as the key of the song.

The reference PCPs for major and minor keys are derived from the major and minor scale. A song of a special key is characterized by the scale of this key. That is if a song is performed in key c the pitches of the scale c namely c d e f g a b c will be the dominant pitches. Pitches that are not part of the scale like c♯ or g♯ are much more unlikely to occur as they produce dissonance. We assume that the key of the input data does not modulate. Thus, the whole song can be used for the calculation of the one key.

Pitch	c	d	e	f	g	a	b
Function	I	II	III	IV	V	VI	VII
Chords Proper to scale	**C**	**d**	**e**	**F**	**G**	**a**	G7 (b-)
Secondary Dominants	**G**	**A**	B	**C**	**D**	**E**	
Secondary Subdominants	**F**	**G**	**A**	**B♭**	**C**	**D**	

Table 1. Selected Chords for Key C-Major

Once the key of the song has been determined the set of possible chords is filtered according to harmonics. The following chords pass this filter: chords that are proper to the scale, except the diminished minor chord on VII. This chord is not included as it is equal to, and usually perceived as the already included dominant sept chord V7 played without its root (E.g. VII for key C-Major consists of the pitches b-d-f, V7 contains the pitches g-b-d-f). Furthermore, all secondary dominants except the dominant of III and all secondary subdominants are selected. Thus, from 24 possible chords (12 major and 12 minor chords) 10 chords are preselected to build the set of possible chords for this song. Table 1 shows the table of chords for key C-Major. Selected chords are formatted bold.

From the 35 test songs 24 keys were correctly identified (68%). The wrongly identified keys were either confusions between corresponding minor and major keys (e.g. C-Major and A-Minor; 9 times) or confusions of a key with its neighbour in the circle of fifth (2 times). Note that confusions between a major and its corresponding minor key do not affect the chord detection algorithm, as both keys are connected to the same set of chords. Also note that confusions of a key with its neighbours in the circle of fifth are not as severe as confusions with keys that are wider away, as in the first case the two keys have more chords in common. Thus neighbouring keys as C-Major and G-Major have 8 out of 10 selected chords in common, whereas the keys with the greatest distance in the circle of fifths, like B♭-Major and E-Major, share only one chord.

3.6. Chord Sequence Optimization

The previous sections described how to obtain for a certain timespan a list of possible chords and their probabilities. These are finally passed to an algorithm that selects from these chords not simply the most probable but the best fitting. This is achieved by analysing the context of each chord. We have obtained the chords at beat time interval, which we declared as the shortest timespan in which a chord change happens. Although chord changes may happen at each beat time they most often last for a longer period of time. Thus, the idea behind this algorithm is to select the chord sequence with high probabilities for each single chord and few chord changes. To this end the probabilities obtained in section 3.3 are combined with a chord change penalty to give the final score.

Total Results	Whole Algorithm	65%
Per Module Evaluation	Core Algorithm	37%
	Core + Beat Module	43%
	Core + Key Detection	50%
	Core + Smoothing	44%

Table 2. Accuracy Rates

4. EVALUATION

In previous research relatively small audio collections were used (2 songs in [4], 8 measures in [5], 7 songs in [7]). Creating test sets for chord detection is laboursome, as the true chord sequences have to be detected manually and each chord has to be assigned to a specific time within the piece of music. Nevertheless, a large and heterogenous test set is desirable as it raises the significance of the evaluation. In order to get representative results a test set of 35 songs of various styles and genres has been assembled and hand-labelled. The spectrum of conteporary test songs ranges from *ABBA's "Dancing Queen"* to *"Californication"* by the *"Red Hot Chili Peppers"*. Furthermore, 3 different interpretations of the *"Second Autumn Allegro"* of *Antonio Vivaldi's "Four Seasons"* and 2 interpretations of the first *Bach Praeludium* have been evaluated. A detailed list of the evaluated songs including the hand-labelled groundtruth files is available online[3].

The effectiveness of each enhancement module has been evaluated against this test set independently of the other modules. Table 2 summarizes the achieved accuracy rates. Accuracy was computed as time in milliseconds where detected and reference chords match, divided by the total length of the song. Each of the modules has raised the average quality by several percent. Overall, the 4 enhancement-modules have improved the chord detection accuracy by 28% compared to a simple short-span detection algorithm. In the final integration test accuracy ranged from 42% to 92%, with an average accuracy rate of 65%.

We identified three main factors that lead to wrong chord identification: The most important factor are non-chord tones: neighbour tones, anticipation, passing tones, suspension and escape tones are just some examples for notes that can occur mainly in the melody, that are not members of the chord and produce intended dissonances. The stronger these tones are, the more they confuse the algorithm. Thus songs where the melody is much more dominant than the accompaniment are more difficult to analyse. For example the Bach Praeludium was once tested in its original version (62%) and once with Gournod's solo melody "Ave Maria" (33.5% accuracy). Other obstacles are the use of chord types which are currently not recognized by our system, like augmented and diminished chords, suspended or extended chords, and misleading frequency spectra due to the harmonic series, mistuning (especially with human voice) or percussive sounds.

5. CONCLUSION

We have introduced a modular chord detection algorithm based on music theory. This algorithm incorporates automatic beat tracking, pitch class profile estimation and key detection from an audio signal, as well as a module for chord sequence smoothing.

We labeled a test set of 35 songs. Though the songs are under copyright restrictions and thus cannot be published, we still hope to facilitate future research by making the corresponding groundtruth files publicly available.

The possibilities offered by music theory are not yet exhausted. Promising enhancements include incorporation of music meter (the chord boundary detection could be enhanced and analysis spans could further be stretched which would make the algorithm more robust against non-chord tones), integration of a chord-sequence database as described in [7], and detection of modulation.

6. REFERENCES

[1] Dixon, S. "Automatic Extraction of Tempo and Beat from Expressive Performances", *Journal of New Music Research*, 30(1): pp. 39–58, 2001

[2] Dixon, S. "Onset Detection Revisited", *Proc. 9th Int. Conf. on Digital Audio Effects (DAFx)*, 2006, pp. 133–137

[3] Harte, C., Sandler, M., Abdallah, S. and Gomez, E. "Symbolic Representation Of Musical Chords: A Proposed Syntax For Text Annotations", *Proc. 4th Int. Conf. on Music Information Retrieval (ISMIR)*, 2005, pp. 66–71

[4] Sheh, A. and Ellis, D.P.W. "Chord segmentation and recognition using EM-trained hidden markov models" *Proc. 4th Int. Conf. on Music Information Retrieval (ISMIR)*, 2003, pp. 183–189

[5] Su, B. and Jeng, S. K. "Multi-timbre chord classification using wavelet transform and self-organized map neural networks" *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001, pp. 3377–3380

[6] Tolonen, T. and Karjalainen, M. "A Computationally Efficient Multipitch Analysis Model" *In IEEE Transactions on Speech and Audio Processing*, 8(6): pp. 208–716, 2000

[7] Yoshioka, Y., Kitahara, T., Komatani, K., Ogata T. and Okuno, H.G. "Automatic Chord Transcription with Concurrent Recognition of Chord Symbols and Boundaries" *Proc. 5th Int. Conf. on Music Information Retrieval (ISMIR)*, 2004, pp. 100–105

[3] http://www.ifs.tuwien.ac.at/mir