REAL-TIME CHORD RECOGNITION FOR LIVE PERFORMANCE

Adam M. Stark, Mark D. Plumbley[*]

Queen Mary University of London
Centre for Digital Music
adam.stark@elec.qmul.ac.uk

ABSTRACT

This paper describes work aimed at creating an efficient, real-time, robust and high performance chord recognition system for use on a single instrument in a live performance context. An improved chroma calculation method is combined with a classification technique based on masking out expected note positions in the chromagram and minimising the residual energy. We demonstrate that our approach can be used to classify a wide range of chords, in real-time, on a frame by frame basis. We present these analysis techniques as externals for Max/MSP.

1. INTRODUCTION

A chord is the simultaneous sounding of two or more musical notes, with the interval relationships between these notes determining the type of chord. The process of automatic chord recognition [6] is one of assigning a chord label to a section of audio. As a number of notes are present, and each note consists of a series of partials, the process typically requires a harmonic analysis of the input signal.

Audio chord recognition has many applications in the area of music information retrieval, such as annotating the harmonic content of audio files in a database or for use in music transcription systems. In this paper we are concerned with the real-time extraction of chord labels from the live performance of a single polyphonic instrument. This information can provide harmonic and structural information about a performance that may be used to increase the ability of computer systems to interact with musicians in a live performance.

A shortfall of previous approaches is the limited number of chords classified. Several systems only consider major and minor triads. This can be a problem if a chord contains more than one triad such as a C Major 7 chord which contains both a C Major triad and an E minor triad. This can lead to misclassifications. However, we wish to design a system capable of classifying audio frames as one of 108 chords, specifically the 12 variations of major, minor, diminished, augmented, suspended 2nd, suspended 4th, major 7th, minor 7th and dominant 7th chords. These are more chords than the majority of approaches in the literature have considered. This is essential if the extraction of chords from real world signals is to be achieved.

In many previous approaches, the first step in developing a chord recognition algorithm has been to convert audio frames into a representation similar to that of the *chroma vector*, also commonly referred to as a *pitch class profile (PCP)* or *chromagram* [4]. A chroma vector is a 12×1 vector with values representing the energy present in each of the 12 semitone pitch classes found in western music.

Several techniques have been used to calculate the chromagram. Some systems [1, 6, 8] make use of the constant-Q transform [2] with some of these approaches employing a tuning algorithm to allow for differences in tuning. Other techniques calculate the chromagram directly from the discrete Fourier transform (DFT) of the input signal by mapping the energy in spectral bins to one of a number of pitch classes [4, 11] . Lee [7] calculates the chroma vector from the result of the Harmonic Product Spectrum of the DFT rather than the DFT itself. Cremer and Derboven [3] present a technique that uses a frequency warped FFT followed by the erasing of overtones and the separation of tonal components from transients. Gomez [5] finds spectral peaks by considering local maxima and using quadratic interpolation to estimate peak magnitudes. The chroma vector is then calculated by weighting each peak by its contribution to each chroma vector bin.

Once the chromagram, or similar representation, has been calculated, a variety of techniques can be used to give it a chord label. Pattern matching techniques generally compare how similar the chroma vector is to a set of chord profiles, usually in the form of bit masks. A bit mask is a 12×1 vector containing a 1 where notes are present and 0 elsewhere. A C Major chord would be represented as [1,0,0,0,1,0,0,1,0,0,0,0].

Two popular pattern matching methods are (i) to find the bit mask with the minimum euclidean distance to the chromagram and (ii) to find the bit mask that maximises the dot product with the chromagram. In the latter approach, a weighting can be used to distinguish between chords containing different numbers of notes. For details of these approaches applied to chord recognition, see [3, 4, 6].

[*] AS is supported by an EPSRC DTA Studentship and MP is supported by an EPSRC Leadership Fellowship.

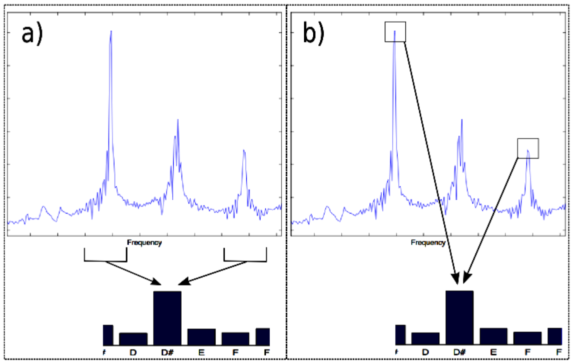

Figure 1. a) The bin mapping technique: energy in spectral bins is mapped to a certain pitch class. b) Our technique: the maximum value in a given range is used as the amplitude value for the harmonic contributing to that pitch class.

Several attempts have been made to classify chords using statistical techniques, in particular hidden Markov Models (HMMs) [1, 8, 10, 11]. Our aim in this paper is to improve baseline extraction in a frame-based classifier for real-time use and so we do not make use of HMMs.

2. APPROACH

2.1. Improved Chromagram Calculation

Some existing chroma calculation techniques, such as the ones used in [4, 11], include all energy within a given frequency range in the amplitude value of a certain pitch class in the chromagram. This idea of using energy within certain spectral ranges was also used by Orio & Schwartz [9] in their Peak Structure Distance (PSD) measure. However, this creates a large potential for unwanted energy such as noise in the chromagram. We therefore wish to develop a technique that identifies only the energy in the harmonics within a given range.

The first step in our approach multiplies the signal frame by a Hamming window and calculates the magnitude spectrum, $X(k)$, using the DFT. The square root of the magnitude spectrum is taken to reduce the amplitude difference between harmonic peaks. Then frequencies of the lower octave, starting from $f_{C3} = 130.81Hz$, are calculated by $f(n) = f_{C3} \cdot 2^{(n/12)}$ for $n = 0, 1, ..., P-1$ where $P = 12$, the number of notes in an octave. An important part of our approach is that we consider only the energy present in the partial itself, by finding the largest peak within a given frequency range, rather than including all energy within that range which can add unwanted energy into our chromagram (see Figure 1). Also, through the process of searching for the partial within a given range, we allow our system to detect partials that are slightly inharmonic. The chroma vector, C, is calculated by:

$$C(n) = \sum_{\phi=1}^{2} \sum_{h=1}^{2} \left(\max_{k_0^{(n,\phi,h)} \leq k \leq k_1^{(n,\phi,h)}} X(k) \right) \left(\frac{1}{h} \right) \qquad (1)$$

where $n = 0, 1, ..., P-1$ where $P = 12$, the number of notes in an octave, ϕ is the number of the octave to consider and h is the number of the harmonic. The value r, for which we choose 2, is the number of bins to search either side of a frequency for a maximum peak, $k_0^{(n,\phi,h)} = k'^{(n,\phi,h)} - (r \cdot h)$, $k_1^{(n,\phi,h)} = k'^{(n,\phi,h)} + (r \cdot h)$ and

$$k'^{(n,\phi,h)} = \text{round}\left(\frac{f(n) \cdot \phi \cdot h}{(f_s/L)} \right) \qquad (2)$$

where f_s is the sampling frequency and L is the frame size.

Through considering the magnitude of spectral peaks our approach has similarities to [5], however as we are focused upon a rhythmic accompaniment, we analyse a much smaller portion of the spectrum. We hypothesise that the majority of instruments playing a rhythmic accompaniment use the lower register of the instrument so our approach examines 2 octaves of the spectrum, between $f_{C3} = 130.81Hz$ and $f_{C5} = 523.25Hz$. Real instruments are not perfectly harmonic and so to reduce problems arising from this only 2 partials are considered and inharmonicity is allowed for by searching for partials within a given range. To be able to comfortably achieve quarter-tone frequency resolution at 130.81Hz we use a sampling frequency of 11025Hz and a frame size of 8192 samples (0.74s).

2.2. Chord Classification

To classify the chroma vector, we employ an approach based upon residual energy in the chromagram. Bit mask representations imply that the energy in a normalised chroma vector for a note that is present will be both close to 1 and similar in value to that of other present notes. A nearest neighbour comparison [4] will yield best results for a chroma vector that has values of exactly 1 for all notes present and 0 otherwise, but our experience has shown that this is unlikely to be the case. The weighted sum technique [4, 6] suffers from a problem when it is necessary to classify chords that contain different numbers of notes. A dot product is likely to produce a larger result for the chord with more notes, so a weighting is needed to differentiate between the chords. Due to the potential for variability in the energy of notes present in the chroma vector, the setting of this weighting is difficult to decide upon and if not thought through carefully, can be arbitrary. As a result we wish to develop a technique that avoids the problem of the variable amplitude of notes.

To solve this problem, we classify chords by masking out the notes hypothesised to be in the chord by each bit mask, instead minimising the energy outside of the mask. This is achieved by finding the minimal dot product between the chromagram and a 'complementary' bit mask, calculated from the original bit mask. To achieve this, for each bit mask, we calculate:

$$\delta_i = \frac{\sqrt{\sum_{n=0}^{P-1} \bar{T}_i(n)(C(n))^2}}{(P-N_i)} \qquad (3)$$

where C is the chroma vector, $\bar{T}_i(n) = 1 - T_i(n)$ where T_i is the ith bit mask, N_i is the number of notes in the ith bit mask and $P = 12$, the number of notes in an octave. Dividing by $(P-N_i)$ is designed to prevent chords with less notes having a natural advantage over other chords. This our equivalent of a 'weighting' between chords but we hypothesise that the amplitude of the noise floor will be more consistent than that of sounded notes and so the potential for error is reduced. We choose the chord that minimises δ_i.

2.3. Chromagram-Unresolvable Chords

Certain chords, when represented using a chromagram, are indistinguishable from other chords as they contain exactly the same notes. From the set of chords that we are attempting to classify, this problem occurs with augmented chords and between some suspended 2nd and 4th chords.

By providing the root note to these chords we can resolve them as well as any other type. However, we do not know the root note and our informal tests have shown that we cannot assume that it will be the lowest note or the note with the most energy. As a solution, we employ the following heuristic.

We extract a 'low' chromagram, C_{low}, from the lowest octave by restricting the value for the octave, ϕ, in equation 1, to 1. We then use this to create a weighting for the suspended chords, W_{sus}, by examining the relationship between the root and fifth (7 semitones higher) in C_{low}:

$$W_{sus}(n) = (1-\alpha)\cdot C_{low}(n) + \alpha \cdot C_{low}(\mathrm{mod}(n+7,P)) \quad (4)$$

where P = 12 and we choose $\alpha = 1/3$. Similarly, we create a weighting for the augmented chords, W_{aug}, by:

$$W_{aug}(n) = (1-\alpha)\cdot C_{low}(n) + \alpha \cdot C_{low}(\mathrm{mod}(n+8,P)). \quad (5)$$

We then apply these weightings to the δ_i values for 'unresolvable' chords so that chords with certain root notes are favoured and consequently these chords become resolvable.

2.4. Compensating for 'Ghost Notes'

Problems can arise relating to overtones of the fundamental which occur at approximately nf_0, for integers $n \geq 2$. Particular problems occur with the 3rd harmonic, $3f_0$, as this is an interval of a fifth above the fundamental and, for some notes, is low enough in frequency to be considered by our algorithm. The result is extra energy in the chroma bin 7 semitones up from each fundamental note.

Unable to solve this problem through changes to the chroma calculation technique, we experimented with the interim solution of adding small bias values, β, to equation 3 that allowed us to correct for the problems introduced by misidentified fundamentals:

$$\delta_i = \frac{\sqrt{\sum_{n=0}^{P-1} \bar{T}_i(n)(C(n))^2}}{(P-N_i)(\beta)}. \qquad (6)$$

We chose $\beta = 1$ as the basic case (no bias) and chose $\beta = 1.06$ for chords that were prone to misidentification due to the presence of these 'ghost notes'. We found that this value did not detract from the ability of the system to correctly identify chords that were not given the bias, it simply reduced the misidentifications to an acceptable number.

3. EVALUATION

The system was implemented offline in Matlab and also as a Max/MSP 'external' for real-time use[1]. A hop size of 1024 samples at 11025Hz was used to achieve more regular chord identification leading to around 10 estimates per second.

We conducted an initial evaluation of our system based upon the classification of audio frames. We created a test set of real world examples of 180 chords played on two different guitars with 4 frames randomly selected from the recording of each chord to create 1440 frames with accompanying labels, totalling over 17 minutes of audio. Each chord type (major, minor, diminished etc) had at least a whole octave in the data set, with some types having up to 2 octaves. We felt that this was a rigourous test method examining the ability of the algorithm to recognise chords over many examples with different root note frequencies. Note that we are focusing upon analysis of a single polyphonic instrument and so our evaluation and results will differ from more generic evaluations on collections of commercial music recordings.

The label of each frame consisted of the *root note* of the chord, the *quality* (major, minor, diminished etc) and any other *intervals* present in the chord (e.g. minor 7th). We recorded the performance of the system at all 3 levels.

We compared our technique to two other chromagram calculation techniques. These were the technique used in [1], an adaptation of the constant-Q based technique presented in [6] which we refer to as *CQ*, and another technique largely related to the techniques used in [4, 11] where spectral bins are mapped to chroma bins which we refer to as *BM*. We shall refer to our technique as the *Proposed* chroma.

As can be seen in Figure 2, our proposed chroma calculation algorithm both performs better at all three levels and has a much lower reduction in performance when all levels are considered. For example, our system identifies the correct root note in 94% of cases and only drops 1.3% when

[1] http://www.elec.qmul.ac.uk/digitalmusic/people/adams/chordrec/

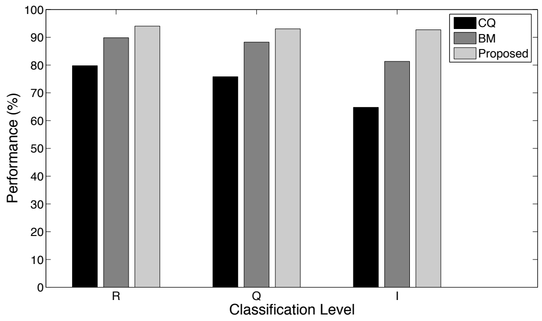

Figure 2. Performance of different chroma calculation techniques. *CQ* = a constant-Q based technique, *BM* = a binmapping technique and *Proposed* = our approach. The results are across all 1440 examples. *R* = Root Note, *Q* = Root Note and Quality and *I* = Root Note, Quality and Intervals.

all three levels are considered. This is in contrast to the CQ technique which suffers a 15% decline in performance.

Examining the results of our method in more detail (Table 1) we find that the system performed very well on most chord types, scoring 92.7% on average for the extraction of the root note, chord quality and any other intervals. Lower performance was achieved on chord types liable to confusion due to identical chromagram representations as discussed in Section 2.3. The system still incorrectly identified some major and minor triads as 7th chords due to 'ghost note' problems identified in Section 2.4. However, this was greatly reduced by our measures to counter this problem.

We also compared our classification technique to other classification techniques. These were a *Nearest Neighbour (NN)* classifier and a *Weighted Sum (WS)*. Our approach showed greater performance by a margin of 1.2% over the NN technique and 20.1% over the WS when root note, chord quality and all other intervals are considered.

Informal tests with several instruments, including a piano and several synthesisers, indicate that our algorithm performs well on those instruments as well as the guitar.

Chord Type	R (%)	Q (%)	I (%)
Major/Minor	99.5	97.9	97.7
Diminished	100	100	100
Augmented	84.0	84.0	84.0
Sus2 / Sus4	85.4	83.3	82.3
Major 7	99.3	99.3	99.3
Minor 7	97.2	97.2	97.2
Dominant 7	100	100	100
Total (Over Examples)	**94.0**	**93.1**	**92.7**

Table 1. Results of evaluation of the chord detection algorithm. *R* = Root Note, *Q* = Root Note and Quality and *I* = Root Note, Quality and Intervals.

4. CONCLUSION

We have presented a real-time chord recognition system that allows for inharmonicity in the input signal and have shown that by classifying chords based upon residual chroma vector energy we can accurately identify many types of chords.

Our experimental results indicate that the system performs very well on real-world guitar recordings, producing improved results over other state of the art approaches. Informal use of the system in a real-time environment appears to confirm this with many different chord types being consistently identified. In future we wish to evaluate the performance of our system across whole performances and to investigate whether the results can be improved using HMMs.

5. REFERENCES

[1] J. P. Bello and J. Pickens, "A robust mid-level representation for harmonic content in music signals," in *Proc ISMIR*, 2005, pp. 304–311.

[2] J. C. Brown, "Calculation of a constant-Q spectral transform," *JASA*, vol. 89, pp. 425–434, January 1991.

[3] M. Cremer and C. Derboven, "A system for harmonic ananlysis of polyphonic music," in *Proc of the AES 25th Int. Conf.*, London, UK, 2004, pp. 115–120.

[4] T. Fujishima, "Real-time chord recognition of musical sound: A system using common Lisp music," in *Proc ICMC*, 1999, pp. 464–467.

[5] E. Gómez, "Tonal description of music audio signals," Ph.D. dissertation, Universitat Pompeu Fabra, 2006.

[6] C. A. Harte and M. B. Sandler, "Automatic chord identification using a quantised chromagram," in *Proc AES 118th Convention*, no. 6412, Barcelona, 2005.

[7] K. Lee, "Automatic chord recognition using enhanced pitch class profile," in *Proc ICMC*, 2006, pp. 306–313.

[8] K. Lee and M. Slaney, "Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio," *IEEE Trans. Audio, Speech Lang. Proc.*, vol. 16, pp. 291–301, 2008.

[9] N. Orio and D. Schwarz, "Alignment of monophonic and polyphonic music to a score," in *Proc ICMC*, 2001, pp. 129–132.

[10] H. Papadopoulos and G. Peeters, "Simultaneous estimation of chord progression and downbeats from an audio file," in *Proc ICASSP*, 2008, pp. 121–124.

[11] A. Sheh and D. P. W. Ellis, "Chord segmentation and recognition using EM-trained hidden Markov models," in *Proc ISMIR*, 2003, pp. 183–189.