

# CHORD RECOGNITION USING INSTRUMENT VOICING CONSTRAINTS

**Xinglin Zhang**

Dept. of Computer Science  
University of Regina  
Regina, SK CANADA S4S 0A2  
zhang46x@cs.uregina.ca

**David Gerhard**

Dept. of Computer Science, Dept. of Music  
University of Regina  
Regina, SK CANADA S4S 0A2  
gerhard@cs.uregina.ca

## ABSTRACT

This paper presents a technique of disambiguation for chord recognition based on *a-priori* knowledge of probabilities of chord voicings in the specific musical medium. The main motivating example is guitar chord recognition, where the physical layout and structure of the instrument, along with human physical and temporal constraints, make certain chord voicings and chord sequences more likely than others. Pitch classes are first extracted using the Pitch Class Profile (PCP) technique, and chords are then recognized using Artificial Neural Networks. The chord information is then analyzed using an array of *voicing vectors* (VV) indicating likelihood for chord voicings based on constraints of the instrument. Chord sequence analysis is used to reinforce accuracy of individual chord estimations. The specific notes of the chord are then inferred by combining the chord information and the best estimated voicing of the chord.

## 1 INTRODUCTION

Automatic chord recognition has been receiving increasing attention in the musical information retrieval community, and many systems have been proposed to address this problem, the majority of which combine signal processing at the low level and machine learning methods at the high level. The goal of a chord recognition system may also be low-level (identify the chord structure at a specific point in the music) or high level (given the chord progression, predict the next chord in a sequence).

### 1.1 Background

Sheh and Ellis [6] claim that by making a direct analogy between the sequences of discrete, non-overlapping chord symbols used to describe a piece of music and word sequence used to describe speech, much of the speech recognition framework in which hidden Markov Models are popular can be used with minimal modification. To represent the features of a chord, they use Pitch Class Profile (PCP) vectors (discussed in Section 1.2) to emphasize the tonal content of the signal, and they show that PCP vectors outperformed cepstral coefficients which are widely used in speech

recognition. To recognize the sequence, hidden Markov Models (HMMs) directly analogous to sub-word models in a speech recognizer are used, and trained by the Expectation Maximization algorithm. Bello and Pickens [1] propose a method for semantically describing harmonic content directly from music signals. Their system yields the Major and Minor triads of a song as a function of beats. They also use PCP as the feature vectors and HMMs as the classifier. They incorporate musical knowledge in initializing the HMM parameters before training, and in the training process. Lee and Slaney [5] build a separate hidden Markov model for each key of the 24 Major/Minor keys. When the feature vectors of a musical piece are presented to the 24 models, the model that has the highest possibility represents the key to that musical piece. The Viterbi algorithm is then used to calculate the sequence of the hidden states, i.e. the chord sequence. They adopt a 6-dimensional feature vector called the Tonal Centroid [4] to detect harmonic changes in musical audio. Gagnon *et al* [2] propose an Automatic Neural Network based pre-classification approach to allow a focused search in the chord recognition stage. The specific case of the 6-string standard guitar is considered. The feature vectors they use are calculated from the Barkhausen Critical Bands Frequency Distribution. They report an overall performance of 94.96% accuracy, however, a fatal drawback of their method is that both the training and test samples are synthetic chords consisting of 12 harmonic sinusoids for each note, lacking the noise and the variation caused by the vibration of the strings where partials might not be in the exact multiple of their fundamental frequencies. Yoshioka *et al* [7] point out that there exists mutual dependency between chord boundary detection and chord symbol identification: it's difficult to detect the chord boundaries correctly prior to knowing the chord; and it's also difficult to identify the chord name before the chord boundary is determined. To solve this mutual dependency problem, they propose a method that recognizes chord boundaries and chord symbols concurrently. PCP vector is used to represent the feature. When a new beat time is examined (Goto's [3] method is used to obtain the beat times), the hypotheses (possible chord sequence candidate) are updated.

## 1.2 Pitch Class Profile (PCP) Vector

A musical note can be characterized as having a global pitch, identified with a note name and an octave (*e.g.* C4) or a pitch color or pitch class, identified only by the note name independent of octave. The pitch class profile (PCP) technique detects the color of a chord based on the relative content of pitch classes. PCP begins from a frequency representation, for example the Fourier transform, then maps the frequency components into the 12 pitch classes. After the frequency components have been calculated, we get the corresponding notes of each frequency component and find its corresponding pitch class.

## 2 CHORD ANALYSIS

Our approach deviates from the approaches presented in Section 1 in several key areas. The use of voicing constraints (described below) is the primary difference, but our low-level analysis is also somewhat different from current work. First, current techniques will often combine PCP with Hidden Markov Models. Our approach analyzes the PCP vector using Neural Networks, using a viterbi algorithm to model chord sequences in time. Second, current techniques normally use window sizes on the order of 1024 samples (23.22 ms). Our technique uses comparatively large window sizes (22050 samples, 500ms). Unlike Gagnon, Larouche and Lefebvre [2], who use synthetic chords to train the network, we use real recordings of chords played on a guitar. Although the constraints and system development are based on guitar music, similar constraints (with different values) may be determined for other ensemble music.

### 2.1 Large Window Segmentation

Guitar music varies widely, but common popular guitar music maintains a tempo of 80–120 beats per minute. Because chord changes typically happen on the beat or on beat fractions, the time between chord onsets is typically 600–750 ms. Segmenting guitar chords is not a difficult problem, since the onset energy is large compared to the release energy of the previous chord, but experimentation has shown that 500ms frames provide sufficient accuracy when applied to guitar chords for a number of reasons. First, if a chord change happens near a frame boundary, the chord will be correctly detected because the majority of the frame is a single pitch class profile. If the chord change happens in the middle of the frame, the chord will be incorrectly identified because contributions from the previous chord will contaminate the reading. However, if sufficient overlap between frames is employed (*e.g.* 75%), then only one in four chord readings will be inaccurate, and the chord sequence rectifier (see Section 2.3) will take care of the erroneous measure: based on the confidence level of chord recognition and

changes in analyzed feature vectors from one frame to the next, the rectifier will select the second-most-likely chord if it fits better with the sequence.

The advantages of the large window size are the accuracy of the pitch class profile analysis, and, combined with the chord sequence rectifier, outweigh the drawbacks of incorrect analysis when a chord boundary is in the middle of a frame. The disadvantage of such a large window is that it makes real-time processing impossible. At best, the system will be able to provide a result half a second after a note is played. Offline processing speed will not be affected, however, and will be comparable to other frame sizes. In our experience, real-time guitar chord detection is not a problem for which there are many real-world applications.

### 2.2 PCP with Neural Networks

We have employed an Artificial Neural Network to analyze and characterize the pitch class profile vector and detect the corresponding chord. A network was first constructed to recognize seven common chords for music in the keys of C and G, for which the target chord classes are [*C, Dm, Em, F, G, Am, D*]). These chords were chosen as common chords for “easy” guitar songs. The network architecture was set up in the following manner: 1 12-cell input layer, 2 10-cell hidden layers, and 1 7-cell output layer. With the encouraging results from this initial problem (described in Section 4), the vocabulary of the system was expanded to Major (I, III, V), Minor (I, iii, V) and Seventh (I, III, V, vii) chords in the seven natural-root(♯) keys (*C, D, E, F, G, A, B*), totaling 21 chords. Full results are presented in Section 4.

A full set of 36 chords (Major, Minor and Seventh for all 12 keys) was not implemented, and we did not include further chord patterns (Sixths, Ninths etc.). Although the expansion from seven chords to 21 chords gives us confidence that our system scales well, additional chords and chord patterns will require further scrutiny. With the multitude of complex and colorful chords available, it is unclear whether it is possible to have a “complete” chord recognition system which uses specific chords as recognition targets, however a limit of 4-note chords would provide a reasonably complete and functional system.

### 2.3 Chord Sequence Rectification

Isolated chord recognition does not take into account the correlation between subsequent chords in a sequence. Given a recognized chord, the likelihood of a subsequent frame having the same chord is increased. Based on such information, we can create a sequence rectifier which corrects some of the isolated recognition errors in a sequence of chords. For each frame, the neural network gives a rank list of the possible chord candidates. From there, we estimate the chord transition possibilities for each scale pair of Major and rel-

ative Minor through a large musical database. The Neural Network classification result is provided in the  $S$  matrix of size  $N \times T$ , where  $N$  is the size of the chord dictionary and  $T$  is the number of frames. Each column gives the chord candidates with ranking values for each frame. The first row of the matrix contains the highest-ranking individual candidates, which, in our experience, are mostly correct identifications by the neural network. Based on the chords thus recognized, we calculate the most likely key for the piece. For the estimated key we develop the chord transition probability matrix  $A$  of size  $N \times N$ . Finally, we calculate the best sequence from  $S$  and  $A$  using the Viterbi Algorithm, which may result in a small number of chord estimations being revised to the second or third row result of  $S$ .

## 2.4 Voicing Constraints

Many chord recognition systems assume a generic chord structure with any note combination as a potential match, or assume a chord “chromaticity,” assuming all chords of a specific root and color are the same chord, as described above. For example, a system allowing any chord combination would identify  $[C-E-G]$  as a  $C$  Major triad, but would identify a unique  $C$  Major triad depending on whether the first note was middle  $C$  ( $C4$ ) or  $C$  above middle  $C$  ( $C5$ ). On the other hand, a system using chromaticity would identify  $[C4-E4-G4]$  as identical to  $[E4-G4-C5]$ , the first voicing<sup>1</sup> of a  $C$  Major triad. Allowing any combination of notes provides too many similar categories which are difficult to disambiguate, and allowing a single category for all versions of a chord does not provide complete information. What is necessary, then, is a compromise which takes into account statistical, musical, and physical constraints for chords.

The goal of our system is to constrain the available chords to the common voicings available to a specific instrument or set of instruments. The experiments that follow concentrate on guitar chords, but the technique would be equally applicable to any instrument or ensemble where there are specific constraints on each note-production component. As an example, consider a SATB choir, with standard typical note ranges, e.g. Soprano from  $C4$  to  $C6$ . Key, musical context, voice constraints and compositional practice means that certain voicings may be more common. It is common compositional practice, for example, to have the Bass singing the root (I), Tenor singing the fifth (V), Alto singing the major third (III) and Soprano doubling the root (I).

This *a-priori* knowledge can be combined with statistical likelihood based on measurement to create a bayesian-type analysis resulting in greater classification accuracy using fewer classification categories. A similar analysis can be performed on any well-constrained ensemble, for example a string quartet, and on any single instrument with multiple

variable sound sources, for example a guitar. At first, the Piano does not seem to benefit from this method, since any combination of notes is possible, and likelihoods are initially equal. However, if one considers musical expectation or human physiology (hand-span, for example), then similar voicing constraints may be applied.

One can argue that knowledge of the ensemble may not be reasonable *a priori* information—will we really know if the music is being played by a wind ensemble or a choir? The assumption of a specific ensemble is a limiting factor, but is not unreasonable: timbre analysis methods can be applied to detect whether or not the music is being played by an ensemble known to the system, and if not, PCP combined with Neural Networks can provide a reasonable chord approximation without voicing or specific note information.

For a chord played by a standard 6-string guitar, we are interested in two features: what chord is it and what voicing of that chord is it<sup>2</sup>. The PCP vector describes the chromaticity of a chord, hence it does not give any information on specific pitches present in the chord. Given knowledge of the relationships between the guitar strings, however, the voicings can be inferred based the *voicing vectors* (VV) in a certain category. VVs are produced by studying and analyzing the physical, musical and statistical constraints on an ensemble. The process was performed manually for the guitar chord recognition system but could be automated based on large annotated musical databases.

Thus the problem can be divided into two steps: determine the category of the chord, then determine the voicing. Chord category is determined using the PCP vector combined with Artificial Neural Networks, as described above. Chord voicings are determined by matching harmonic partials in the original waveform to a set of context-sensitive templates.

## 3 GUITAR CHORD RECOGNITION SYSTEM

The general chord recognition ideas presented above have been implemented here for guitar chords. Figure 1 provides a flowchart for the system. The feature extractor provides two feature vectors: a PCP vector which is fed to the input layer of the neural net, and an voicing vector which is fed to the voicing detector.

Table 1 gives an example of the set of chord voicing arrays and the way they are used for analysis. The fundamental frequency ( $f_0$ ) of the root note is presented along with the  $f_0$  for higher strings as multiples of the root  $f_0$ .

The Guitar has a note range from  $E2$  (82.41Hz, open low string) to  $C6$  (1046.5Hz, 20th fret on the highest string). Guitar chords that is above the 10th fret ( $D$ ) are rare, thus we can restrict the chord position to be lower than the 10th

<sup>1</sup> A voicing is a chord form where the root is somewhere other than the lowest note of the chord

<sup>2</sup> Although different voicings are available on guitar, a reasonable assumption is that they are augmented with a root bass on the lowest string

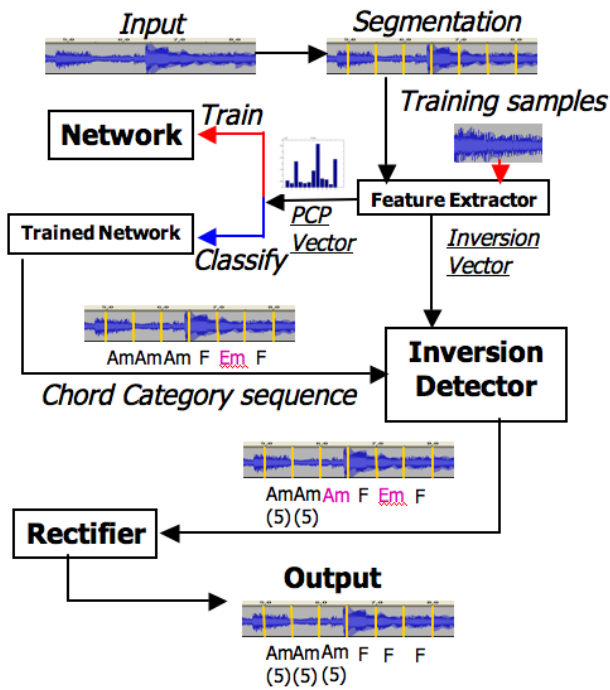


Figure 1. Flowchart for the chord recognition system.

fret, that is, the highest note would be 10th fret on the top string, i.e. *D*<sub>5</sub>, with a frequency of 587.3Hz. Thus if we only consider the frequency components lower than 600Hz, the effect of the high harmonic partials would be eliminated.

Each chord entry in Table 1 provides both the frequencies and first harmonics of each note. “Standard” chords such as Major, Minor and Seventh, contain notes for which  $f_0$  is equal to the frequency of harmonic partials of lower notes, providing consonance and a sense of harmonic relationship. This is often be seen as a liability, since complete harmonic series are obscured by overlap from harmonically related notes, but our system takes advantage of this by observing that a specific pattern of harmonic partials equates directly to a specific chord voicing. Table 1 shows this by detailing the pattern of string frequencies and first harmonic partials. First harmonic partials above *G*<sub>6</sub> are ignored since they will not interact with higher notes. Harmonic partials above 600Hz are ignored, since there is no possibility to overlap the fundamental frequency of higher notes (as described above). These are indicated by the symbol “ $\emptyset$ ”. In this way, we construct a pattern of components that are expected to be present in a specific chord as played on the guitar. A string that is not played in the chord is indicated by “–”. Boxes and underlines are detailed below. Table 2 shows the same information for voicings of a single chord in three different positions, showing how these chords can be disambiguated.

Chord	S1	S2	S3	S4	S5	S6
$f_0$ (Hz)	H1	H2	H3	H4	H5	H6
F	1	1.5	2	2.52	3	4
87.31	2	3	4	$\emptyset$	$\emptyset$	$\emptyset$
Fm	1	1.5	2	<u>2.38</u>	3	4
87.31	2	3	4	$\emptyset$	$\emptyset$	$\emptyset$
F7	1	1.5	<u>1.78</u>	2.52	3	4
87.31	<span style="border: 1px solid black;">2</span>	3	<span style="border: 1px solid black;">3.56</span>	$\emptyset$	$\emptyset$	$\emptyset$
G	1	1.26	1.5	2	2.52	4
98	2	2.52	<span style="border: 1px solid black;">3</span>	4	$\emptyset$	$\emptyset$
Gm	1	<u>1.19</u>	1.5	2	3	4
98	2	<span style="border: 1px solid black;">2.38</span>	3	4	$\emptyset$	$\emptyset$
G7	1	1.26	1.5	2	2.52	<u>3.56</u>
98	2	2.52	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
A	–	1	1.5	2	2.52	3
110	–	2	3	$\emptyset$	$\emptyset$	$\emptyset$
Am	–	1	1.5	2	<u>2.38</u>	3
110	–	2	3	$\emptyset$	$\emptyset$	$\emptyset$
A7	–	1	1.5	<u>1.78</u>	2.52	3
110	–	<span style="border: 1px solid black;">2</span>	3	$\emptyset$	$\emptyset$	$\emptyset$
C	–	1	1.26	1.5	2	2.52
130.8	–	2	2.52	$\emptyset$	$\emptyset$	$\emptyset$
Cm	–	1	<u>1.19</u>	1.5	2	–
130.8	–	2	$\emptyset$	$\emptyset$	$\emptyset$	–
C7	–	1	1.26	<u>1.78</u>	2	2.52
130.8	–	2	2.52	$\emptyset$	$\emptyset$	$\emptyset$
D	–	–	1	1.5	2	2.52
146.8	–	–	2	$\emptyset$	$\emptyset$	$\emptyset$
Dm	–	–	1	1.5	2	<u>2.38</u>
146.8	–	–	2	$\emptyset$	$\emptyset$	$\emptyset$
D7	–	–	1	1.5	<u>1.78</u>	2.52
146.8	–	–	<span style="border: 1px solid black;">2</span>	$\emptyset$	$\emptyset$	$\emptyset$

Table 1. Chord pattern array, including three forms of five of the natural-root chords in their first positions. S1–S6 are the relative  $f_0$  of the notes from the lowest to highest string, and H1–H6 are the first harmonic partial of those notes. See text for further explanation of boxes and symbols.

Chord	S1	S2	S3	S4	S5	S6
$f_0$ (Hz)	H1	H2	H3	H4	H5	H6
G	1	1.26	1.5	2	2.52	4
98	2	2.52	<span style="border: 1px solid black;">3</span>	4	$\emptyset$	$\emptyset$
G(3)	1	1.5	2	2.52	3	4
98	2	3	4	$\emptyset$	$\emptyset$	$\emptyset$
G(10)	–	1	1.5	2	2.52	3
196.0	–	2	3	$\emptyset$	$\emptyset$	$\emptyset$

Table 2. Voicing array for the *Gmaj* chords played on different positions on the guitar.

### 3.1 Harmonic Coefficients and Exceptions

It can be seen from Table 1 that there are three main categories of chords on the guitar, based on the frequency of the second note in the chord. The patterns for the three categories are: (1.5), where the second note is (V): *F, Fm, F7, E, Em, E7, A, Am, A7, B, Bm, D, Dm, D7*; (1.26), where the second note is (III): *B7, C7, G, G7*; and (1.19), where the second note is (iii): *Cm, Gm*.

Thus, from the first coefficient (the ratio of the first harmonic peak to the second) we can identify which group a certain chord belongs to. After identifying the group, we can use other coefficients to distinguish the particular chord. In some situations (e.g., *F* and *E*; *A* and *B*), the coefficients are identical for all notes in the chord, thus they cannot be distinguished in this manner. Here, the chord result will be disambiguated based on the result of the Neural Network and the  $f_0$  analysis of the root note. Usually, all first harmonic partials line up with  $f_0$  of higher notes in the chord. When the first harmonic falls between  $f_0$  of higher notes in the chord, they are indicated by boxed coefficients.

Underlined coefficients correspond to values which may be used in the unique identification of chords. In these cases, there are common notes within a generic chord pattern, for example the root (1) and the fifth (1.5). String frequencies corresponding to the Minor Third (1.19, 2.38) and Minor Seventh (2.78) are the single unique identifiers between chord categories in many cases.

## 4 RESULTS

Chord detection errors do not all have the same level of “severity”. A *C* chord may be recognized as an *Am* chord (the relative Minor), since many of the harmonic partials are the same and they share two notes. In many musical situations, although the *Am* chord is incorrect, it will not produce dissonance if played with a *C* chord. Mistaking an *C* chord for a *Cm* chord, however, is a significant problem. Although the chords again differ only by one note, the note in question is more harmonically relevant and differs in more harmonic partials. Further, it establishes the mode of the scale being used, and, if played at the same time as the opposing mode, will produce dissonance.

### 4.1 Chord Pickout

*Chord Pickout*<sup>3</sup> is a popular off-the-shelf chord recognition system. Although the algorithm used in the *Chord Pickout* system is not described in detail by the authors, it is reasonable to compare with our system since *Chord Pickout* is a commercial system with good reviews. We applied the same recordings to both systems and identified the accuracy of each system. We were more forgiving with the

analysis for *Chord Pickout* in order to better detail the types of errors that were made. If *Chord Pickout* was able to identify the root of the chord, ignoring Major, Minor or Seventh, it is described as “correct root.” If the chord and the chord type are both correct, it is described as “correct chord.” Errors between correct root and correct chord included Major→Minor, Major→Seventh, and Minor→Major. For our system, all chord errors regardless of severity, are considered incorrect. The complete results for 6 trials are presented in Table 3.

### 4.2 Independent accuracy trials

To detect the overall accuracy of our system, independent of a comparison with another system, we presented a set of 40 chords of each type to the system and evaluated its recognition accuracy. Two systems were trained for specific subsets of chord detection.

The first system was trained to detect chords in the a single key only assuming key recognition has already taken place. Seven chord varieties are available as classification targets, and the system performed well, producing 96.8% accuracy over all trials. Misclassifications were normally toward adjacent chords in the scale.

The second system was trained to recognize Major, Minor and Seventh chords of all seven natural-root keys, resulting in 21 chord classification targets. This system produced good results: 89% for Major versus Minor, and 75% accuracy for Major versus Seventh chords. Table 4 provides a confusion matrix between single-instance classification of Major and Seventh chords, which had the lower recognition rate. There are two reasons for this: in some cases the first three notes (and correspondingly the first three harmonic partials detected) are the same between a chord and its corresponding Seventh; and in some cases the first harmonic of the root note does not line up with an octave and thus contributes to the confusion of the algorithm. Recognition accuracy is highest when only the first two notes are the same (as in *C* and *G* chords). Recognition accuracy is low in the case of *F7*, when the root is not doubled, and the pattern can be confused with both the corresponding Major chord and the adjacent Seventh chord. Recognition accuracy is also low in the case of *G7*, where the difference between the Major and the Seventh is in the third octave, at 3.56 times the fundamental of the chord. In this case, the Seventh chord is frequently mistaken for the Major chord, which can be considered a less “severe” error since the Seventh chord is not musically dissonant with the Major chord.

A more severe case is with *D7*, which contains only 4 sounded strings, one of which produces a harmonic that does not correspond to a higher played string. From Table 1, we can see that the string frequency pattern for *D7* is [1, 1.5, 1.78, 2.25], and the first harmonic partial of the root note inserts a 2 into the sequence, producing [1, 1.5, 1.78,

<sup>3</sup><http://www.chordpickout.com>

Trial	Frames	Voicing Constraints		Chord Pickout			
		Correct	Rate	Correct Root	Rate	Correct Chord	Rate
1	24	23	95.8%	20	83.3%	1	5.0%
2	46	44	95.6%	30	65.2%	12	26.1%
3	59	54	91.5%	38	64.4%	7	11.9%
4	50	49	98.0%	31	62.0%	30	60.0%
5	65	51	78.4%	51	78.4%	21	32.3%

**Table 3.** Comparison of our system to *Chord Pickout*, an off-the-shelf chord recognition system.

Chord	Rate	C	C7	D	D7	E	E7	F	F7	G	G7	A	A7
C	40/40	40											
C7	35/40		35		2	2						1	
D	40/40			40									
D7	13/40			1	13		1	3	20				2
E	40/40					40							
E7	37/40						37				3		
F	38/40					1	1	38					
F7	5/40					3	16	16	5				
G	40/40									40			
G7	17/40				2					21	17		
A	30/40			1	8							30	1
A7	25/40	5	2								8		25

**Table 4.** Confusion Matrix for Major and Seventh chords of natural-root keys. Overall accuracy is 75%.

2, 2.25]. This is very similar to the sequence for *F7*, which is why the patterns are confused. It would be beneficial, in this case, to increase the weight ascribed to the fundamental frequency when the number of strings played is small. Unfortunately, detecting the number of sounded strings in a chord is a difficult task. Instead,  $f_0$  disambiguation can be applied when a chord with fewer strings is one of the top candidates from the table, since that information is known.

## 5 CONCLUSIONS

A chord detection system is presented which makes use of voicing constraints to increase accuracy of chord and chord sequence identification. Although the system is developed for guitar chords specifically, similar analysis could be performed to apply these techniques to other constrained ensembles such as choirs or string, wind, or brass ensembles, where specific chords are more likely to appear in a particular voicing given the constraints of the group.

## 6 REFERENCES

- [1] J. P. Bello and J. Pickens. A robust mid-level representation for harmonic content in music signals. In *Proceedings of the International Symposium on Music Information Retrieval*, London, UK, 2005.
- [2] T. Ganon, S. Larouche, and R. Lefebvre. A neural network approach for preclassification in musical chord recognition. In *37th Asilomar Conf. Signals, Systems and Computers*, volume 2, pages 2106–2109, 2003.
- [3] M. Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2):159–171, 2001.
- [4] C. Harte, M. Sandler, and M. Gasser. Detecting harmonic change in musical audio. In *AMCMM '06: Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, 2006.
- [5] K. Lee and M. Slaney. A unified system for chord transcription and key extraction using hidden markov models. In *Proceedings of International Conference on Music Information Retrieval*, 2007.
- [6] A. Sheh and D. P. Ellis. Chord segmentation and recognition using em-trained hidden markov models. In *Proceedings of the International Symposium on Music Information Retrieval*, Baltimore, MD, 2003.
- [7] T. Yoshioka, T. Kitahara, K. Komatani, T. Ogata, and H. Okuno. Automatic chord transcription with concurrent recognition of chord symbols and boundaries. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, 2004.