Chord analysis using ensemble constraints

David Gerhard and Xinglin Zhang

Abstract Many applications in music information retrieval require the analysis of the harmonic structure of a music piece. In Western music, the harmonic structure can be often be well illustrated by the chord structure and sequence. This chapter presents a technique of disambiguation for chord recognition based on *a priori* knowledge of probabilities of voicings of the chord in a 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, and make some impossible. Pitch classes are extracted, and chords are then recognized using pattern recognition techniques. The chord information is then analyzed using an array of voicing vectors 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

Traditional Western music is performed by instruments, including the human voice, and instruments are constrained. All the instruments have their characteristic ranges, timbres, playing styles and techniques. Each instrument (including voice parts) has a standard *range* of notes that are playable. For example, the modern piano has a total of 88 keys, ranging from A0 to C8. Instruments also have corresponding

David Gerhard
Department of Computer Science, University of Regina, Regina, SK CANADA e-mail: david.gerhard@uregina.ca

Xinglin Zhang
Department of Computer Science, University of Regina, Regina, SK CANADA e-mail: xinglinzh@gmail.com

standard playing techniques, which can often be derived from the physical structure and layout of the instruments as well as the physical abilities of human player, and are sometimes related to the style of music being played. For example, the guitar is commonly played in one of two styles: chording, where a group of strings is sounded simultaneously, and picking, where individual notes are played to create a melody.

1.1 Prescriptive and Descriptive Constraints

A distinction should be made between *prescriptive* and *descriptive* constraints. Prescriptive means that there is a rule prescribing a musical constraint. The rule can be about composition, technique or any other aspect of the musical characteristics of a piece, but what makes a rule prescriptive is that it can be broken. It is a common practice, put in place by earlier composers and it suggests ways of making music which "sounds good". An example is the prescription to avoid the motion of parallel fifths. Composers have found that motion of parallel fifths can be distracting, can lead to a reduced sense of the perception of the key or chord pattern of the piece, and can make musical critics unhappy. The prescriptive constraint comes from real or imagined reasons for not using a particular construct, but there is no physical reason a composer cannot use motion of parallel fifths. A renegade composer needing just that distraction or wanting to upset critics in this way, is not physically prevented from using parallel fifths. It is a convention, not a requirement. The term "prescribe" means "to write before" and prescriptive constraints are rules which are written before the composer begins to write the song.

A descriptive constraint, on the other hand, is a physical condition of the instrument, ensemble or player which is being asked to produce sounds. And some constraints are so rigid that they cannot be broken. For example, no matter how convincing the request is, a clarinet can never produce more than one note. No matter what enticements or threats are brought to bear, a trumpet cannot play a C two octaves below middle C. And no human can play a chord containing notes that are five octaves apart using a single hand on a piano. Descriptive constraints detail characteristics of the music which cannot be changed usually. There are other descriptive constraints that can be broken by using some certain techniques. For example, the musical *range* of an instrument is an example of a descriptive constraint. This is a standard measure of the notes that an instrument can play, and are so rigid that they are often programmed into musical composition software[1]. However, it is possible, for example, to use some overtone techniques to produce higher pitches that are theoretically "out of range" for a guitar. Another example is on an alto saxophone, where it is possible to achieve a note lower than concert D♭3 by partially blocking the bell against the player's leg, both causing a resonance node and slightly elongating the effective pipe length, producing a lower pitch. The term "describe" means

[1] For example, the *Sibelius* composing software (`http://www.sibelius.com/`) colors a note red when it is out of range for the staff instrument.

"to write down" and descriptive constraints are used to describe the instrument itself and the "standard" playing techniques ascribed to that instrument.

In the context of music information retrieval, prescriptive and descriptive constraints can both be used as additional information to enhance analysis and disambiguate results. Developers must take care when using these constraints, because while prescriptive constraints yield an increased or decreased probability of occurrence from the norm, depending on the rule, rigid descriptive constraints theoretically lead to musical events or characteristics with zero probability. In practice, however, it should be noted that musicians and composers are, by nature, creative, and often seek ways to overcome constraints both prescriptive and descriptive, as mentioned above. The piano, however, has no such techniques and the playable range (A0 to C8) is a true descriptive constraint.

1.2 Application of Constraints in Music Information Retrieval Research

Although these constraints limit the extent of the sounds which can be produced, they can be used by researchers studying music to narrow down the possible answers to the questions they are asking. Researchers have made use of prescriptive and descriptive constraints in music information retrieval in the past, but the distinction between the types of constraints is rarely made, and estimated probability distribution measures are often used in place of identified constraints. Automatic generation of musical instrument fingerings is a research area in which algorithms are used to calculate an optimal fingerings (hand positions and movements) for a sequence of given notes, either monophonic or polyphonic.

Tuohy and Potter [20] present a genetic algorithm for the automatic generation of playable guitar tablature through the use of a fitness function that assesses the playability of a given set of fretboard positions. Though not explicitly using the term "constraints", their fitness function takes into account some physical characteristics of the guitar. "Hand movement" and "Hand manipulation" are considered by the fitness function, which favors easier situations and penalizes complicated situations. In [9], a system for generation of piano fingerings is proposed. The system constrains fingerings by requiring that the same finger be used as long as a note persists and no finger substitution is allowed[2]; each finger may only depress one note. What's more, they use "vertical cost", "horizontal cost" and "user specification of cost function" to measure the playability of the underlying fingering. The three costs take into consideration different kinds of constraints.

Radicioni *et al* [18] explicitly use the term "constraints-based approach" in their system for modeling guitar performers' gestures and annotating a musical score. They believe that the physical gestures used to operate musical instruments are responsible for the characteristics of the sound being produced in a performance. They

[2] Finger substitutions happen in reality for long notes

make use of the highly constrained nature of performers' gestures to build a model of music performance, coupled with a strategy aimed at maximizing the gestural comfort of performers. They draw on physical and bio-mechanical constraints for their model, from the implications of the fact that notes have certain positions on the guitar fretboard and human's hands have a certain range of span, making certain fingerings more frequent than others.

When the music score is known *a priori*, an interesting problem is to discover the fingerings for automatic performance environments [2], learning aid systems [13, 22] and so on. These approaches make use of constraints to achieve this goal. In the situation that we do not know the music scores, *i.e.* that we only have the recorded audio signal, we are interested in discovering both the score and the fingering information. If a system already exists that can discover fingerings based on a score, one can concentrate on the transcription of music signals into scores which can then be passed to the fingering system. Since the audio signals contain sounds which are produced by instruments according to a score (written or not) using specific fingerings which provide us with useful information for the possible combination of notes, we can use the fingering information to assist our transcription. Moreover, fingerings have a constrained nature, thus a constraint-based approach can be applied to chord detection.

This chapter is based around a detailed example of constraint-based chord analysis and tracking, using descriptive constraints based on chordal strumming technique for a standard-tuned guitar with six strings (E4, B3, G3, D3, A2, E2), considering the physical layout of the instrument along with human physical and temporal constrains. The following subsection introduces current research on chord detection in general. Several standard chord-detection techniques will be employed in our constraint-based approach.

1.3 Chord Detection

The harmonic structure of a musical work depicts the content of the music in a high level, illustrating how the music is organized. Many applications in music information retrieval such as semantic analysis of music, finding similar music and segmentation into characteristic parts, require the harmonic structure as a mid-level representation of the musical piece under analysis. The *chord*, which is defined as several notes played simultaneously, is used to represent the harmonic structure in the form of a sequence, or temporal arrangement, of chords depicting the overall structure of the music. Thus the recognition of chords plays an important role in music information retrieval. Many researchers have expended great effort on the

chord detection task, and many MIREX[3] tasks begin with chordal analysis. Audio chord detection was a separate task in the 2008 MIREX competition.

The most popular method used for chord detection is a pattern classification approach, which first extracts low-level features describing harmonic content, *e.g.*, pitch class profile (PCP) vectors (introduced in Sect. 2), or, in other words chroma vectors, then uses a classifier such as a hidden Markov model (HMM) to perform the recognition. PCP and HMM techniques are described in more detail in Sect. 2.

Ryynänen and Klapuri [19] follow the general method, but instead of one PCP vector, they use two-pitch class profiles, one for low-register notes D1–C♯3 and one for high-register notes D3–C♯5. They argue that the low-register profile captures the bass notes contributing to the chord root whereas the high-register profile has more clear peaks for the major/minor third and the fifth. Instead of estimating chord profiles for all chords, they estimate the profiles only for major and minor triads to prevent the problem of insufficient training data for particular chords. A chord HMM with 24 states is defined, twelve states each for both major and minor triads[4]. The observation likelihoods for each chord are calculated by comparing the low and high-register profiles with the estimated trained profiles. Then Viterbi decoding through the chord HMM produces chord labeling for each analysis frame.

Ellis and Poliner [4] models the distribution of the chroma vector using a single Gaussian Model. They make use of beat tracking, and extract the chroma vector one per beat. Because chords usually change at the beginning of a beat, features extracted using beat boundaries are believed to be more confident. They also use an ergodic[5] HMM with states corresponding to chord labels. It is worth noting that chord changes on beat boundaries is a prescriptive constraint, rather than a descriptive constraint, however, it is a very common composition practice in Western music.

Instead of using an ergodic chord HMM, where the hidden states represent chords and all possible transitions between states are allowed, Khadkevich and Omologo [10] create a separate model for each chord. Their approach also differs from others in that they use 512 Gaussian mixtures (a weighted sum of guassian distributions) representing the chroma vector probabilities rather than the standard 12. In order to prevent difficulties from lack of training data, similar to [19], they also only train 2 models: a major profile and a minor profile. The parameters of the HMMs are obtained using expectation maximization, which iteratively estimates the maximum likelihood estimates of parameters in a probabilistic model.

Weil and Durrieu [23] add a preprocessing step which attenuates the main melody of the musical piece. They believe that the main melody often contains intentionally anharmonic notes. This is an example of breaking a prescriptive constraint, that of ensuring melody notes fall within the notes of the underlying chord structure.

---

[3] The Music Information Retrieval Evaluation eXchange (`http://www.music-ir.org/mirex/2008`) is a competition between researchers to compare the accuracy of algorithms written to solve common music information retrieval problems. It takes places as a part of the International Conference on Music Information Retrieval (ISMIR).

[4] According to the Audio Chord Detection task in MIREX20008, the chord vocabulary for the task is restricted to 12 major triads (Cmaj, C♯maj,...,Bmaj) and 12 minor triads (Cm,C♯m,...,Bm)

[5] not sensitive to initial conditions

While these anharmonic notes are crucial for the perceived richness of the global timbre, they also blur the accompanying harmonies and make chord detection difficult, which is why they attempt to remove these notes with pre-processing. To make the chroma vectors robust, they also estimate the offset of the tuning frequency relative to A4=440Hz[6]. In this system, they use a system of tonal centroid vectors [8], where a 6-D vector is obtained from the 12-D PCP vector as features, and an ergodic HMM is trained.

While the abovementioned systems have to be trained to get the HMM parameters, Papadopoulos and Peeters [14, 15] derive the HMM parameters manually, taking into account the presence of high harmonics of pitch notes and some music knowledge. In this approach, no training is needed. They represent each chord profile as a vector which contains the theoretical amplitude values of the notes and their harmonics comprising a specific chord. The observation possibilities are then obtained by computing the correlation between the observation vectors and a set of chord profiles. An ergodic HMM is also used.

Pauwels *et al* [16] use a novel feature extractor which first uses multiple pitch tracking techniques to couple the higher harmonics to their fundamental frequency and then compute the chroma vector from these harmonics. Different from [19], they require that the bass-notes with fundamental frequencies lower than 100Hz are not allowed to contribute to the chroma vector. They argue that although bass-notes could make a significant addition to the chord, which agrees with [19], bass notes are often duplicates of notes in the higher register, or they do not contribute to the chord (*e.g.* a walking bass).

There are many other works that also have great contribution to this problem. Uchiyama *et al* [21] use pre-processing step which eliminates percussive sounds from audio, because percussive sounds are non-harmonic and they interfere with chord detection. Lee [11, 12] builds key-dependent HMMs for chord transcription and key extraction, using an HMM for each of the 24 keys, thus detecting the key and chord sequence concurrently. Bello and Pickens [1] present a system for detecting harmonic content in music signals, using chroma vectors and HMMs.

Although some researchers are not explicitly using the term constraint, it is clear that constraints are a common theme in chord detection research. Playability constraints, physical layout constraints for both the instruments and human abilities, and stylistic constraints have appeared in the literature. The "standard" model for chord detection, that of feature extraction using a chroma-type technique combined with some form of pattern classification system, makes several assumptions that many researchers use because they are considered standard. In the following sections, we will describe a new approach in detail, including many of the low-level details and why certain decisions are appropriate in this domain. Before that, there are a few common terms in the chord detection research field that should be explained.

[6] A4 or Concert A is the 440 Hz tone that serves as the standard for musical pitch. A4 is the musical note A above middle C (C4)

2 Term Explanation

Several concepts in the area of chord recognition are well known to researchers immersed in the work but may not be as familiar to occasional readers. We present here a detailed description of some of these concepts. Readers familiar with chord recognition and pattern analysis techniques may be inclined to bypass this section.

2.1 Pitch Class Profile (PCP) vector

A played note gives us many perceptual properties: one is pitch, which corresponds to the frequency of the note: the higher the frequency, the higher the pitch. The second property, related to pitch, is the note name regardless of octave, and corresponds to one of the twelve pitch classes. This property can be termed the "color" of the note. Although chords are made up of notes perceived as absolute pitches, the sensation of a chord is more often that of a coherent sound, with the color property of the notes being dominant over the pitch. Thus, "color" can be used as a term to represent a chord based on its root, corresponding to one of the twelve pitch classes, *i.e.* C, C♯/D♭, D, D♯/E♭, E, F, F♯/G♭, G, G♯/A♭, A, A♯/B♭, B. If we know the power distribution of a chord in each of the twelve pitch classes, which can be represented as a twelve-dimensional vector, the color of a chord can be quantitatively represented. This vector is called the chroma vector or Pitch Class Profile (PCP) vector, which maps the notes in several octaves into 12 bins of pitch classes. The PCP vector technique was first proposed by Fujishima [5] in 1999 for the representation of audio and it is widely used today to represent the features of a chord for analysis and classification.

2.2 Artificial Neural Networks

An artificial neural network (ANN), often just called neural network for short, is a mathematical model simulating biological networks of neurons. It is composed of a group of interconnected processing elements. Each such "neuron" is in fact a function, taking some input and producing an output based on the input, usually as a type of summative threshold function. A nerual network can be used for pattern classification [3] through a learning process. The most common type of ANN used for pattern classification is a three-layer feedforward network. Feedforward means connections between units do not form a directed cycle. The information moves in only one direction, forward from the input layer to the output layer through the hidden layer. Although a complete description of ANNs is beyond the scope of this work, the interested reader will find a complete description in many pattern recognition or information theory textbooks, including [3].

2.3 Hidden Markov Models and the Viterbi algorithm

Hidden Markov models are the pattern recognition technique most commonly used for detecting patterns in temporal sequences. The sequence of chords in a musical piece is an example of a temporal sequence. HMMs work by building a model of an underlying system the states of which are unobserved (hidden) but which produces a series of observations based on the internal hidden states. HMMs are used for classification by lining up an observed sequence with the possible observed sequences generated by the model. Often, a number of HMMs are compared and the one most likely to have produced the observed sequence is judged to best model the internal structure of the system being classified.

The Viterbi algorithm is a standard process for finding the sequence of hidden states with the highest likelihood in a particular HMM. Given a set of hidden states and transition probabilities between them, the Viterbi algorithm proceeds by finding the most likely state at each time increment, and maintaining a list of the most likely historical sequences of states that would lead to the current state. Again, a complete description of both HMMs and the Viterbi algorithm is beyond the scope of this work, and interested readers are encouraged to consult [17] for details.

3 Chord Analysis in Guitar music

The remainder of the chapter will present, in detail, the analysis and classification of guitar chords using constraints based on the physical construction of the instrument. The chords in the experiments are played in an acoustic guitar. We call this technique "voicing constraints" because it can identify different chord voicings based on constraints of the instrument and chord voicings which might be less easy to play or even impossible.

Our approach deviates from the approaches presented in the introduction section in several key areas. The use of voicing constraints (described below) is the primary and fundamental 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 since Neural Networks are a more stable method and are more capable of capturing the probability distribution of the PCP vectors than a single Gaussian or a Gaussian Mixture Model (GMM) and using a pseudo-Viterbi algorithm to model chord sequences in time. Second, current techniques normally use small window size. Our technique makes use of comparatively large window sizes (500ms). The description and justification of these methods is presented in the following sections. Although the constraints and system development are based on guitar music, similar constraints (with different values) may be determined for other ensemble music.

3.1 Large Window Segmentation

Choosing the size of the analysis window for feature extraction is always a challenge. Small windows are able to localize higher-frequency events in time, while larger windows can localize longer-time events in frequency. Depending on the application, a larger or smaller window may be appropriate, but it should be noted that no window size is appropriate for all applications. A type of "uncertainty principle" exists between time-localization and frequency-localization: in order to know the exact instant when an event takes place, down to the sample, one cannot know anything about the frequency content of the event, since the event consists of a single sample. Likewise, in order to fully analyze the frequency content of an event, the event must be of infinite length[7]. In chord recognition, some constraints can be identified which will allow us to use a larger window than most researchers do, and therefore produce a more detailed frequency analysis.

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 (a prescriptive constraint as mentioned earlier), and because of physical limitations of the way guitar chords are played, we can see that statistically, 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. The chord pitch class profile pattern does not change from onset to onset (although local relative changes will occur), even taking into account effects such as slides and hammer-ons, since those would produce small but measurable onsets themselves. Because of this, the entire signal from one onset to the next can be taken as a single frame when calculating the pitch class. This results in a more accurate pitch class analysis than for small window sizes. Further, using a larger window like this has the effect of blurring the analysis of percussive (fast, high-frequency) events, which makes them easier to ignore.

Experimentation has shown that onset detection, while a useful addition to the algorithm, is not entirely necessary. Universal 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, as shown in Fig. 1. 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 Sect. 3.4) will take care of the erroneous measure.

The advantage of the large window size is the accuracy of the pitch class profile analysis, and, combined with the chord sequence rectifier, this outweighs the possible 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 process-

[7] Theoretically, Fourier analysis requires an infinite-length signal, however, in practice, we add 0 outside.

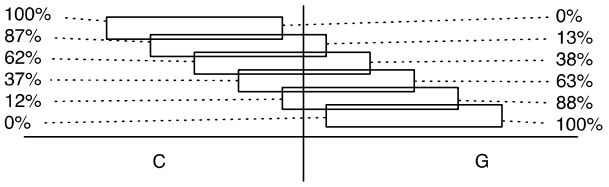

Fig. 1 Large frames at 75% overlap across a chord boundary. Frames that cross the boundary will either be dominated by one chord and successfully recognized, or contain similar contributions from both chords and possibly be mis-recognized.

ing 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.

3.2 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]). It is common practice for chord detection systems under development to begin with this or a similar subset of all possible chords. These chords were chosen as common chords for "easy" guitar songs. The network architecture was set up in the following manner:

- 1 input layer, 2 hidden layers, and 1 output layer;
- 12 input cells corresponding to 12 elements the of PCP vector;
- 10 cells for each of the hidden layer;
- 7 output cells corresponding to the 7 chosen chords.

With the encouraging results from this initial problem (described in Sect. 4), the vocabulary of the system was expanded to recognize chords in the seven roots (C, D, E, F, G, A, B). The system was trained to recognize Major (I, III, V), Minor (I, iii, V) and Seventh (I, III, V, vii) chords for each key, totaling 21 chords. Recognition rates were lower than with the seven-chord system, as may be expected, but still very good. The complete set of major, minor and seventh chords for all 12 chord roots would include 36 chords. 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[8].

[8] Only Major, Minor and Seventh Chords are considered in this system.

3.3 Supervised Training with Real-world Data

Unlike Gagnon, Larouche and Lefebvre [6], who use synthetic chords to train the network, we use real recordings of chords played on a guitar. In the research discussed in the introduction section which adopts HMMs as the classifier, the training material is a labelled chord sequence, including many chords in one training sample, typically a song with the chord labels. However, our system is trained using separate chords, *i.e.* one training sample contains only one chord. A guitar is connected to the audio input of the computer, and chords are recorded using Audacity[9] , an open source audio editor. These recorded chords are then labeled and fed into the system as training data.

3.4 Sequence Rectifier

Chord recognition rarely operates on a single isolated chord. This is primarily a prescriptive constraint, since there is no physical reason why any chord can't follow any other. The only descriptive aspect of this constraint is the time it takes to move from one chord to another. Chord transitions that are faster that the human hand can move would be disallowed. Some chord transitions are easier to make, for example, while maintaining a common finger position, and some transitions are unlikely to happen, for example briefly switching to a relative minor chord and then to the subdominant chord. These constraints, combined with a measure of the certainty of the initial chord detection result, can increase the recognition accuracy of individual chords.

After the chord detection system is trained, it can be used to classify the chord for each frame. Because the frame length is usually smaller than the time interval during which a chord is played, we are likely to have several instances of the same chord recognized in a sequence. For example, if a C chord is played for 1 second, and assuming a 500ms frame overlapped at 75% as justified above, the correct output should contain 6 instances of the C chord. If a chord boundary falls in the middle of one of the frames, or if noise or other difficulties are present, the network might erroneously produce an Am chord (for example) instead of a C chord. Based on the confidence level of the chord recognition as well as changes in analyzed feature vectors from one frame to the next, we construct a *sequence rectifier* which will select the second-most-likely chord if it fits better into the sequence. In this way, the rectifier improves the overall accuracy of the system.

For each frame, the neural network gives a rank-ordered list of the possible chord candidates, each with a confidence value in the range of [0 1]. The sequence rectifier algorithm is:

1. Estimate the chord transition possibilities for each key pair (Major and relative minor) through large musical database.

[9] http://audacity.sourceforge.net/

2. The Neural Network provides a matrix S, which has N rows and T columns. Each column gives the chord candidates with ranking values for each frame. N is the size of the chord dictionary. T is the number of frames.
3. Based on the first row of matrix S, calculate the most probable key for the entire piece of music. For the 24 possible keys, the key corresponding to the maximum number of the chords in the first row of S wins. This is an example of *key finding*, another common MIREX task.
4. Using the estimated key, construct the transition matrix A from step 1.
5. Calculate the best sequence from S and A using the Viterbi Algorithm.

3.5 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" (see Sect. 2.1) are the same chord. For example, a system like this would identify [C4-E4-G4] as identical to [E4-G4-C5], the first inversion[10] of the C Major triad. Although these chords have the same chromaticity (all contain C, E, and G components), they will sound different to the ear. A system which not only identifies [C-E-G] as a C Major triad, but also can identify a unique C Major triad depending on whether the first note was middle C (C4) or C above middle C (C5), would be preferable in most circumstances.

For a chord, allowing *any* combination of notes regardless of the *voicings*[11] 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 since equivalent chords in different octaves are not disambiguated. What is necessary, then, is a compromise which takes into account statistical, musical, and physical likelihood constraints for chord patterns.

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-producing component. As an example, consider a SATB choir, with typical note ranges as shown in Table 1

In this example, then, some chord voicings are more likely than others, depending on the key of the piece, the chord progression, and the melodic movement. Further, compositional practice (a prescriptive constraint) means that depending on the musical context, certain voicings may be more common, for example, it is common compositional practice to have the Bass singing the root (I), Tenor singing the fifth

[10] An *inversion* of a chord is an arrangement of notes where the triad begins with the root (root position), the third (first inversion), or the fifth(second inversion)

[11] A chord *voicing* is a specific way of arranging the notes which make up the chord. An inversion is a special case of a voicing.

Table 1 Typical note ranges for SATB choir.

Voice	Range
Soprano	C4-C6
Alto	E3-E5
Tenor	C3-C5
Bass	C2-C4

(V), Alto singing the third (III or iii) and Soprano doubling the root (I) when the chord being sung is the root of the key.

This *a priori* knowledge can be combined with statistical likelihood based on measurement to create a Bayesian 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 piano. 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 and human physiology (hand-span, for example), then similar voicing constraints may be constructed.

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: metadata may tell us the instrumentation, and 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 being used. 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 (both prescriptive and descriptive) on an ensemble. Here, 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, and then determine the voicing. The chord category is determined using the PCP vector combined with Artificial Neural Networks, as described previously. Chord voicings are determined by matching harmonic partials in the original waveform (extracted using Fourier or Constant-Q transforms, for example) to a set of voicing templates.

When the chord is strummed, it is possible that not all the strings are sounded. For example, we may strum the first 4 strings or the middle 3 strings in a chord. Because it is impossible to identify which strings may be missing in a particular

chord, we must take into account that the VVs against which we are matching may be missing one or more feature values. This kind of problem can be described as pattern recognition with incomplete feature vectors. Standard methods are available for this type of problem.

4 Guitar Chord Recognition System

The general chord recognition ideas presented above have been implemented here for guitar chords. Figure 2 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 a voicing vector which is fed to the voicing detector. The rectifier corrects the errors, which are marked in purple.

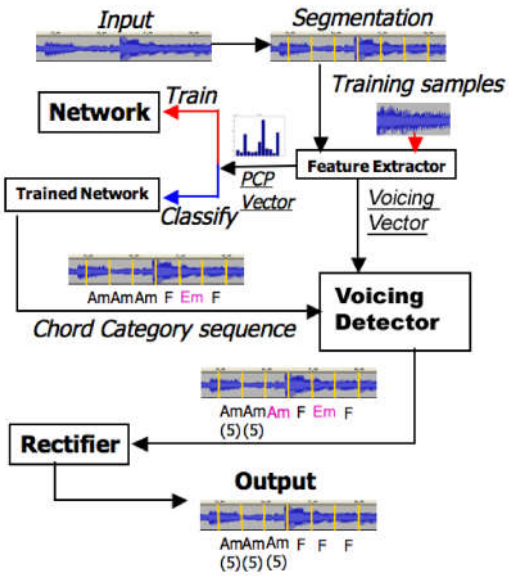

Fig. 2 Flowchart of the chord recognition system. Note: the yellow lines showing segmentation do not show overlap, but overlap is done as shown in Fig. 1.

Table 2 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 are above the 10th fret of which the highest note is D5 are rare, thus we can restrict the chord position to be lower

Chord f_0, in Hz	S1 H1	S2 H2	S3 H3	S4 H4	S5 H5	S6 H6
E	1	1.5	2	2.52	3	4
82.4	2	3	4	ϕ	ϕ	ϕ
Em	1	1.5	2	2.38	3	4
82.4	2	3	4	ϕ	ϕ	ϕ
E7	1	1.5	1.78	2.52	3	4
82.4	2	3	3.56	ϕ	ϕ	ϕ
F	1	1.5	2	2.52	3	4
87.31	2	3	4	ϕ	ϕ	ϕ
Fm	1	1.5	2	2.38	3	4
87.31	2	3	4	ϕ	ϕ	ϕ
F7	1	1.5	1.78	2.52	3	4
87.31	2	3	3.56	ϕ	ϕ	ϕ
G	1	1.26	1.50	2	2.52	4
98	2	2.52	3	4	ϕ	ϕ
Gm	1	1.19	1.50	2	3	4
98	2	2.38	3	4	ϕ	ϕ
G7	1	1.26	1.50	2	2.52	3.56
98	2	2.52	ϕ	ϕ	ϕ	ϕ
A	–	1	1.5	2	2.52	3
110		2	3	ϕ	ϕ	ϕ
Am	–	1	1.5	2	2.38	3
110		2	3	ϕ	ϕ	ϕ
A7	–	1	1.5	1.78	2.52	3
110		2	3	ϕ	ϕ	ϕ
B	–	1	1.5	2	2.52	3
123.5		2	3	ϕ	ϕ	ϕ
Bm	–	1	1.5	2	2.38	3
123.5		2	3	ϕ	ϕ	ϕ
B7	–	1	1.26	1.78	2	3
123.5		2	2.52	ϕ	ϕ	ϕ
C	–	1	1.26	1.50	2	2.52
130.8		2	2.52	ϕ	ϕ	ϕ
Cm	–	1	1.19	1.5	2	–
130.8		2	ϕ	ϕ	ϕ	ϕ
C7	–	1	1.26	1.78	2	2.52
130.8		2	2.52	ϕ	ϕ	ϕ
D	–	–	1	1.5	2	2.52
146.8			2	ϕ	ϕ	ϕ
Dm	–	–	1	1.5	2	2.38
146.8			2	ϕ	ϕ	ϕ
D7	–	–	1	1.5	1.78	2.52
146.8			2	ϕ	ϕ	ϕ

Table 2 Chord pattern array, including three forms of five of the natural-root chords in their first voicings (root always the lowest). 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. For example, in the E chord, String 2 plays the ≈123.6Hz B ($1.5 \times f_0$), and the first harmonic is twice that($3 \times f_0$). See text for further explanation of boxes and symbols.

than the 10th fret, that is, the highest note would be 10th fret on the top string, *i.e.* D5, 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. And because the guitar has only six strings, if the six strings are all strummed, we only have 6 fundamental frequencies. The relationship between fundamental frequencies of the notes in the chord can be used to identify the voicing.

Each chord entry in Table 2 provides both the frequencies of all sounded notes on each string (or an indication that the string is not played) as well as an indication of the first harmonic 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. Indeed, this is why these chords are pleasant to hear. This can be seen as a liability, since complete harmonic series are obscured by overlap from harmonically related notes. Current systems attempt to overcome this by reinforcing, re-interpolating or duplicating harmonics, but our system takes advantage of this by observing that a specific pattern of harmonic partials equates directly to a specific voicing of a chord. Table 2 shows this by detailing the pattern of string frequencies and first harmonic partials for the root voicings of these chords. 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 "ϕ" In this way, we construct a pattern of components that are expected to be present in a specific chord as played on the guitar, and similarly for other voicings.

4.1 Harmonic Coefficients and Exceptions

If we make an assumption that the lowest sounded note on the guitar is the root of the chord (which is prescriptive but not unreasonable), it can be seen from Table 2 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
- (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 the chords themselves 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 $\boxed{\text{boxed}}$ 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.

4.2 Feature Extractor

The feature extractor in Fig. 2 includes two parts: a PCP extractor and a voicing extractor.

4.2.1 PCP Extractor

As introduced in Sect. 2, a PCP vector has 12 dimensions, corresponding to the 12 pitch classes. The value for each dimension represents the energy in the corresponding pitch class, and this value is usually normalized with the largest value equal to 1. PCP begins from a frequency representation of the audio of a chord, for example, in our implementation, we use the fast Fourier transform (FFT). To calculate the PCP vector, first we use FFT to get the frequency components of the frame and then map the frequency components into the 12-bin pitch classes. After the frequency components have been calculated, we get the corresponding notes of each frequency component and find its corresponding pitch class. Then we add the power of each frequency component to the corresponding pitch class. In this way, PCP provides a profile of the frequency components in an audio signal, regardless of octave.

4.2.2 Voicing Extractor

The voicing extractor uses FFT to get the frequency components for each frame, and then a peak finding algorithm[12] is used to find the most evident peaks from the frequency components that are lower than 600Hz. After the peak frequencies are obtained, the lowest frequency among them is selected as the bass note and then the voicing vector is obtained by dividing all the frequency component by the bass frequency.

[12] http://terpconnect.umd.edu/ toh/spectrum/PeakFindingandMeasurement.htm

5 Accuracy and Obstacles

In this section we describe the accuracy of this implementation of a chord constraints system, taking into account the relative severity of common chord errors, and we compare this system to an off-the-shelf system.

5.1 Common Chord Errors

It is important to recognize that chord detection errors do not all have the same level of what can be called "severity". A major chord (*e.g.* C) may be recognized as the relative minor of the same chord (*e.g.* Am) since they are based around the same set of accidentals, and many of the harmonic partials are the same because 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. This can be seen as analogous to the problem of octave errors in pitch detection, where the pitch is incorrect but would not produce dissonance if played with the correct note. Relative Minor chords are perceptually more similar to a Major chord than is the corresponding same-root Minor chord. Mistaking an F chord for an Fm chord, for example, is a significant problem. Although the chords again differ only by one note, the note in question 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.

5.2 Comparison

Chord Pickout[13] 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 developers, it is not unreasonable to make a comparison 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". Inconsistencies between the correct root and correct chord include Major-Minor, Major-Seventh, and Minor-Major confusions.

We also make a comparison with *CLAM Chorddata*[14], which implements a chord detector using the algorithm proposed by Christopher Harte [7]. For each frame,

[13] `http://www.chordpickout.com/`

[14] `http://clam.iua.upf.edu/index.html`

CLAM gives several chord candidates. If the correct chord is contained in the candidates, we regard this as a correct recognition.

For our system, all chord errors are treated as incorrect, regardless of severity. The complete results for 5 trials are presented in Table 3. The results of the inversion constraints are of the same order of magnitude as other modern techniques, which is encouraging but not particularly impressive. It is important to keep in mind that this new system is also able to determine specific voicings of a chord, as will be described below. The main result is that we are able to identify these voicings while maintaining overall chord detection accuracy.

		Inversion Constraints		CLAM		Chord Pickout			
Trial	Frames	Correct	Rate	Correct	Rate	Root	Rate	Chord	Rate
1	281	255	90.8%	240	85.4%	190	67.6%	42	14.9%
2	322	286	88.8%	301	93.4%	172	53.4%	72	22.4%
3	405	356	88.0%	368	90.8%	225	55.6%	56	13.8%
4	466	396	84.9%	410	87.9%	293	62.9%	50	10.7%
5	472	387	81.9%	403	85.3%	321	68.0%	101	21.4%

Table 3 Comparison of our Inversion Constraints system to *CLAM Music Annotator* and *Chord Pickout*.

5.3 Independent Accuracy Trials

To evaluate the overall accuracy of our system, independent of a comparison with other systems, we presented a set of chord exemplars (50 of each type) to the system and evaluated its recognition accuracy. Two systems were trained for specific subsets of chord detection, and the results are presented in three tables. The first set of results, presented in Table 4 shows the recognition accuracy of a system trained to detect chords appropriate to the key of C and D, as discussed above. The system used seven chord classification targets, and produced 93.2% accuracy over all trials. Misclassifications in this case were normally toward adjacent chords in the scale.

Chord	C	Dm	Em	F	G	Am	D
Rate	50/50	48/50	43/50	50/50	48/50	42/50	45/50

Table 4 Recognition results for common chords. Overall accuracy is 93.2%.

The second system was trained to recognize Major, Minor and Seventh chords of all seven natural-root keys, resulting in 21 chord classification targets. Classification results are presented for Major versus Minor comparisons as shown in Table 5, which produce good results (86.8% accuracy). The most common errors were between a Major chord and its relative Minor, although errors between the Major chord

and the same-root Minor were also detected. Table 6 provides a confusion matrix for chord recognition between Major and Seventh chords. The accuracy is reduced in these results, for two reasons: 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. Also, in some cases the first harmonic of the root note does not line up with the fundamental frequency of a note an octave above, and thus contributes to the confusion of the algorithm.

Chord	C	Cm	D	Dm	E	Em
Rate	50/50	39/50	48/50	41/50	46/50	38/50

Chord	F	Fm	G	Gm	A	Am
Rate	47/50	42/50	50/50	36/50	49/50	35/50

Table 5 Recognition results for natural-root Major and Minor chords. Overall accuracy is 86.8%

		chord was recognized as:													
Chord	Rate	C	C7	D	D7	E	E7	F	F7	G	G7	A	A7	B	B7
C	50/50	50													
C7	35/50	12	35												3
D	50/50			50											
D7	26/50			14	26			3	6		1				
E	45/50					45	2	3							
E7	26/50					9	26	3	12						
F	48/50					1	1	48							
F7	32/50					2	2	14	32						
G	50/50									50					
G7	35/50									15	35				
A	48/50											48	2		
A7	31/50	2										17	31		
B	45/50							2						45	3
B7	29/50		10											11	29

Table 6 Confusion Matrix for Major and Seventh chords of natural-root keys. Overall accuracy is 78.6%.

Recognition accuracy is higher for the Major chords and lower for Seventh chords. Taking E7 for example, there are 9 samples where the E7 chord is recognized as E Major. Examining Table 2, the voicing vectors for E Major and E7 are [1, 1.5, 2, 2.52, 3, 4] and [1, 1.5, 1.78, 2.52, 3, 4]. Since the first element in E7 produces a harmonic at twice the fundamental, the detected vector is [1, 1.5, 1.78, 2, 2.52, 3, 4]. If the third element 1.78 is too weak for the peak picking algorithm to detect, we may erroneously detect [1, 1.5, 2, 2.52, 3, 4] instead, which is exactly the voicing vector for E major. Since E Major and E7 have the same root, we recognize E7 as an E Major in this situation.

For the B7 chord, in addition to recognizing it as the corresponding Major chord, 20% are recognized as C7, one semitone above B7. The voicing vector for B7 is [1, 1.26, 1.78, 2, 3] in which the second element 1.26 produces a second harmonic peak at 2.52, leading to the array [1, 1.26, 1.78, 2, 2.52, 3], the first five elements of which are the same as that of C7. Moreover, since there a small difference (130.8Hz − 123.5Hz = 7.3Hz) between the root f_0 of B7 and C7, if the guitar is tuned slightly higher, B7 might be recognized wrongly as C7. And we notice the same error happens for E7, where the difference between the frequencies of the roots of E7 and F7 is 4.91Hz. This indicates that correct tuning as well as correct peak identification are significant problems for chord recognition, and will manifest more obviously in chords which differ by a semitone (B-C and E-F) than chords which differ by a tone.

Another difficult case is with D7, which contains only 4 notes (the two lowest notes are not sounded for D chords in the root position), and the first note produces a harmonic that does not correspond to a higher played string. From Table 2, we can see that the string f_0 multiplier pattern for D7 is [1, 1.5, 1.78, 2.52], and the first harmonic partial of the root note inserts a 2 into the sequence, producing [1, 1.5, 1.78, 2, 2.52] for the sequence of harmonics within the range [82.4Hz–600Hz] as previously justified. 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, if that information is known. Further, the confusion matrix adds to the measure of certainty of the chord recognition, indicating when other methods should be employed to increase confidence.

5.4 Recognition of Voicings

After the chord labels are recognized, our system then gives the voicing information. For the guitar chords, different voicings correspond to different hand positions on the guitar neck. Figure 3 shows 3 voicings of the C Major chord and 3 of the F Major chord, which correspond to positions that are usually used by guitar players. An empty circle means an open string while a filled circle means the string is pressed on the corresponding fret. Numbers in the left-top represent the fret. For example, "'5"' denotes the 5th fret. If there is no number, it means the first fret by default.

It is reasonable to assume that all the strings involved in the chord are played, but sometimes the bass note might be ignored. Thus we present 2 voicing vectors for a particular voicing, the first one including all the strings and the second one with the bass note missed. For voicing C1, the first voicing vector is [1 1.33 1.68 2 2.67 3.36] with a bass frequency 98 Hz. The second voicing vector is [1.33 1.68 2 2.67 3.36]. To keep a constant no. of elements in the voicing vector, we add a sixth element "4", which is caused by the third harmonic of 1.33 and the second

harmonic of 2. Normalizing it, we get [1 1.26 1.5 2.0 2.52 3] with the bass frequency 98×1.33=130.8 Hz. Table 7 shows the voicing vectors for the C Major and F Major Voicings.

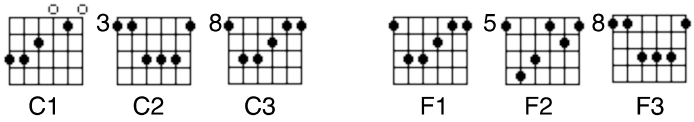

Fig. 3 Three different possible voicings of the C Major and F Major chords.

Voicing	Bass (Hz)	Voicing Vectors					
C1	98	1	1.33	1.68	2	2.67	3.36
	130.8	1	1.26	1.5	2	2.52	3
C2	98	1	1.33	2	2.67	3.36	4
	130.8	1	1.5	2	2.52	3	4
C3	130.8	1	1.5	2	2.52	3	4
	196	1	1.33	1.68	2	2.67	3
F1	87.3	1	1.5	2	2.52	3	4
	130.8	1	1.33	1.68	2	2.67	3
F2	110	1	1.59	2	2.38	3.17	4
	174.6	1	1.26	1.5	2	2.52	3
F3	130.8	1	1.33	2	2.68	3.36	4
	174.6	1	1.5	2	2.52	3	4

Table 7 Voicing vectors For C Major voicings and F Major voicings, including root-played and root-missing.

A total of 55 recorded samples played by 3 persons for each voicing were recorded and tested, the strumming style was used and they were not told whether to play the bass note or not. Table 8 and Table 9 show the confusion matrix for the disambiguation of the voicings for C Major and F Major respectively. The system compares the detected voicing vector with the vectors in Table 7 and recommends the most likely voicings. The sum of each row in the confusion matrices is larger than the number of testing samples because the system can recommend more than one voicing in certain circumstances. Table 8 shows that the system is able to recommend the right voicings at a rate of $(48+55+55)/3 \times 55$=95.7%, though there are extra recommendations. There are many circumstances where C2 is recognized as C1 because of the similarity between the second vector of these voicings and the same bass notes. Also, many C3 chords are recognized as C2 because the first vector

for C3 is the same as the second vector for C2. Table 9 shows that the system works better for disambiguating F Major voicings, with a correct recommendation rate of $(51+55+55)/(3\times 55)$=97.6% and fewer overlapping incorrect recommendations.

	recognized as:		
Voicing	C1	C2	C3
C1	48	15	1
C2	45	55	24
C3	0	52	55

Table 8 Confusion Matrix for Disambiguating C Major Voicings. Multiple recognitions are possible.

	recognized as:		
Voicing	F1	F2	F3
F1	51	4	30
F2	0	55	0
F3	2	5	55

Table 9 Confusion Matrix for Disambiguating F Major Voicings. Multiple recognitions are possible.

6 Conclusions

When performing music information retrieval on recorded audio, applying constraints can greatly reduce the number of candidate recognition results which must be considered. Such constraints can be either prescriptive or descriptive. Descriptive constraints usually indicate physical constraints of the player or instrument which are quite unlikely to be broken. Prescriptive constraints usually indicate stylistic or cultural choices which can be broken but are statistically relevant.

Chord analysis in particular can benefit from applying physical constraints. Not every chord can be played by a specific polyphonic instrument or monophonic instrument ensemble. Depending on the key, style, instrumentation and other parameters, certain chord candidates can be discarded.

Current chord analysis techniques often disregard specific note information in favor of a chord color or, in other words pitch class profile technique. Pitch class profiles cannot disambiguate between inversions or voicing of a single chord, nor can they identify where in the musical range a chord may have been played. By enumerating possible and common chord voicings, it is possible to improve standard Pitch class profile techniques by identifying the chord voicing.

These techniques are demonstrated in a chord detection system we developed 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.

References

1. J. P. Bello and J. Pickens. A robust mid-level representation for harmonic content in music signals. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR-05)*, London, UK., September 2005.
2. G. Cabral, I. Zanforlin, R. Lima, H. Santana, and G. Ramalho. Playing along with d'Accord guitar. In *Proceedings of the 8th Brazilian Symposium on Computer Music*, 2001.
3. Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.
4. D. P. W. Ellis and G. E. Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2007*, volume 4, pages IV–1429–IV–1432, 15–20 April 2007.
5. T. Fujishima. Real-time chord recognition of musical sound: A system using Common Lisp Music. In *Proceedings of the International Computer Music Conference*, pages 464–467, Beijing, China, 1999.
6. T. Gagnon, S. Larouche, and R. Lefebvre. A neural network approach for preclassification in musical chords recognition. In *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 2106–2109, 9–12 Nov. 2003.
7. C. A. Harte and M. Sandler. Automatic chord identification using a quantised chromagram. *Proc. of the 118th Convention. of the AES*, 2005.
8. Christopher Harte, Mark Sandler, and Martin Gasser. Detecting harmonic change in musical audio. In *AMCMM '06: Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pages 21–26, New York, NY, USA, 2006. ACM.
9. A. Kasimi, E. Nichols, and C. Raphael. A simple algorithm for automatic generation of polyphonic piano fingerings. In *Proceedings of the 8th International Conference on Music Information Retrieval*, pages 355–356, 2007.
10. Maksim Khadkevich and Maurizio Omologo. Mirex audio chord detection (abstract). In *the Music Information Retrieval Exchange*. [online (accessed January 2008): `http://www.music-ir.org/mirex/2008/abs/khadkevich_omologo_final.pdf`], 2008.
11. K. Lee. *A System for Acoustic Chord Transcription and Key Extraction from Audio Using Hidden Markov Models Trained on Synthesized Audio*. PhD thesis, Stanford University, Mar. 2008.
12. Kyogu Lee and M. Slaney. Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):291–301, Feb. 2008.
13. Masanobu Miura, Isao Hirota, Nobuhiko Hama, and Masazo Yanagida. Constructing a system for finger-position determination and tablature generation for playing melodies on guitars. *Syst. Comput. Japan*, 35(6):10–19, 2004.
14. H. Papadopoulos and G. Peeters. Large-scale study of chord estimation algorithms based on chroma representation and HMM. In *Proceedings of the International Workshop on Content-Based Multimedia Indexing CBMI '07*, pages 53–60, 25–27 June 2007.
15. H. Papadopoulos and G. Peeters. Simultaneous estimation of chord progression and downbeats from an audio file. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2008*, pages 121–124, March 31 2008–April 4 2008.

16. Johan Pauwels, Matthias Varewyck, and Jean-Pierre Martens. Audio chord extraction using a probabilistic model (abstract). In *the Music Information Retrieval Exchange*. [online (accessed January 2008): http://www.music-ir.org/mirex/2008/abs/mirex2008-audio_chord_detection-ghent_university-johan_pauwels.pdf], 2008.
17. Lawrence Rabiner. A tutorial on HMM and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.
18. Daniele P. Radicioni and Vincenzo Lombardo. A constraint-based approach for annotating music scores with gestural information. *Constraints*, 12(4):405–428, 2007.
19. Matti P. Ryynänen and Anssi P. Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Comput. Music J.*, 32(3):72–86, 2008.
20. D. R. Tuohy and W. D. Potter. A genetic algorithm for the automatic generation of playable guitar tablature. In *Proceedings of the International Computer Music Conference*, pages 499–502, Barcelona, Spain, Sept. 2005.
21. Yuki Uchiyama, Kenichi Miyamoto, and Shigeki Sagayama. Automatic chord detection using harmonic sound emphasized chroma from musical acoustic signal (abstract). In *the Music Information Retrieval Exchange*. [online (accessed January 2008): http://www.music-ir.org/mirex/2008/abs/khadkevich_omologo_final.pdf], 2008.
22. A. B. Viana, J. H. F. Cavalcanti, and P. J. Alsina. Intelligent system for piano fingering learning aid. In *Proceedings of the Fifth International Conference on Control, Automation, Robotics & Vision (ICARCV-98)*, 1998.
23. Jan Weil and Jean-Louis Durrieu. An HMM-based audio chord detection system: Attenuating the main melody (abstract). In *the Music Information Retrieval Exchange*. [online (accessed January 2008): http://www.music-ir.org/mirex/2008/abs/Mirex08_AudioChordDetection_Weil_Durrieu.pdf], 2008.