Technology Reports
Chord Recognition Performance Evaluation Chroma Vector

Musical Chord Recognition Technique, and Applications Thereof

Service Innovation Department Housei Matsuoka Mizuki Watabe

Due to recent advances in artificial intelligence, a growing number of practical systems are employing technologies such as voice interaction. At NTT DOCOMO, by incorporating absolute pitch and musical intelligence into voice interaction systems of this sort, we have developed acoustic recognition technology that can evaluate musical performances with the aim of realizing an agent that can understand music. This technology can recognize musical chord progressions and evaluate ad-lib performances. In this article, we describe our chord recognition technology and an ad-lib performance evaluation function that uses it.

1. Introduction

The effects of music on living beings and their state of mind have been researched in a wide range of fields including physiology, psychology, medicine, nursing and music therapy, and some reports have even included scientific data on effects such as relaxation and stress relief [1] [2].

It has been shown that playing music stimulates the brain, and could play an important role in the treatment of conditions such as dementia in the aging society of the future.

At NTT DOCOMO, we have therefore developed acoustic recognition technology for the assessment of musical performances with the aim of developing an agent that helps people to enjoy giving musical performances even when alone. We focused our attention on a technique for recognizing chords in music, and used a Deep Neural Network (DNN)*1 to improve the chord recognition accuracy.

We also applied this chord recognition technique to the development of an ad-lib performance evaluation function. With this function, the user plays a favorite recording and ad-libs on a musical

*1 DNN: A machine learning method based on the use of a neural network with multiple intermediate layers.

instrument while listening to the recording. An intelligent agent determines whether or not the ad-lib performance matches the recorded music, and responds accordingly. This results in an agent that can understand the user's performance, and could help the user to enjoy playing music or become more motivated to practice.

This article presents an overview of chroma vectors as a means of understanding the pitch of musical instruments, and discusses our technique for extracting these chroma vectors. We also describe the recognition method used in our chord recognition technique, a recognition method that uses a DNN, and we report on the accuracy of these methods. We also describe an ad-lib evaluation function that uses this chord recognition technique.

2. Chroma Vectors

2.1 Overview

In the recognition of musical pitch, a single note can easily be recognized by using an f0 extraction*2 technique (e.g., autocorrelation*3). However, for a polyphonic instrument like a guitar or a piano, f0 extraction is difficult because it is sometimes only possible to recognize a single pitch, and sometimes the incorrect pitch is recognized. To analyze sounds and chords that include multiple pitches, feature values*4 called chroma vectors are often used [3].

A chroma vector is a vector that has feature quantities representing the amplitude of oscillations at the frequency of each of the 12 notes of a musical scale spanning multiple octaves. The amplitude intensity indicates how many signals of a specific frequency are contained in a certain audio segment. The frequencies of each musical scale are fixed. For example, the note A has a frequency of 440 Hz, and the note E has a frequency of 660 Hz. Doubling the frequency produces a note one octave higher, and halving the frequency produces a note one octave lower. Since musical pitch is determined by the ratios of frequencies in this way, the notes of the 12-tone musical scale increase in frequency by a ratio of $1\sqrt[12]{2}$ per semitone. A chroma vector is obtained by calculating the amplitude intensity of each frequency in each scale. This shows which notes of the scale are loudest, making it possible to estimate the chord that is playing based on the combination of strong notes.

2.2 Calculating Chroma Vectors

1) Using Fourier Transforms

To determine the amplitude intensity of a certain musical scale, we have to calculate the average value of the amplitude intensity in the vicinity of these frequencies of the musical scale. A Fourier transform*5 is often used to extract frequency components from sound (signals). Frequency analysis based on Fourier transforms is performed using orthogonal frequencies. If T is the time duration of one of the segments to be analyzed, then the intervals between these orthogonal frequencies is $1/T$ (Hz), so to increase the frequency resolution, the time duration of the segments to be analyzed has to be increased to some extent. In particular, in the low pitch region, the frequency difference between one note and the next is very small, so the Fourier transforms must have a finer frequency resolution than the frequency difference between neighboring pitches. This makes it difficult to analyze fast melodies.

*2 f0 extraction: The analysis of the lowest reference frequency in a waveform signal that includes overtones.

*3 Autocorrelation: A technique where a signal is correlated with itself after applying a varying temporal offset. The offset timing at which a strong correlation is obtained is inferred to be the signal's reference frequency.

*4 Feature values: Values extracted from data, and given to that data to give it features.

*5 Fourier transform: A process that extracts the frequency components making up a signal and their respective ratios.

Figure 1 (a) shows the results of frequency analysis by Fourier transform. If the duration of a time frame is T seconds and the number of samples per transform is n, then the Fourier transform has base frequencies*6 of $1/T$, $2/T$, $3/T$, ..., n/T. The number of samples is the number of discrete quantized parts that a single time segment is divided into. If we want to analyze frequencies down to a low A (55 Hz), the next highest pitch is B♭ (58 Hz), so we required a frequency resolution such that $1/T$ is less than 3 Hz. This means that T has to be at least 333 ms, making it impossible to analyze melodies where the pitch changes faster than 333 ms.

2) Using Our Technique

Fig. 1 (b) shows the results of frequency analysis using our technique. In this technique, the fundamental frequencies in the Fourier transform are set to musical pitch frequencies, and the amplitude intensities are calculated by the following formula in the same way as for a Fourier transform.

$$p(f) = \sum_{k=0}^{n} \cos\left(\frac{2\pi f k}{SF}\right) - i \sum_{k=0}^{n} \sin\left(\frac{2\pi f k}{SF}\right) \quad (1)$$

SF is the sampling frequency, $P(f)$ is the amplitude and phase information of frequency f of the musical scale, and the square of $P(f)$ is the amplitude intensity of frequency f. The sampling frequency indicates how many times the signal waveform is sampled per second. The cosine term obtains the correlation with a cosine wave, and the sine term obtains the correlation with a sine wave. Unlike an ordinary Fourier transform, not all pairs of frequencies satisfy the orthogonality condition, so there is the drawback that they can interfere with one another. However, it is still possible to analyze the frequency components of each musical scale, facilitating the extraction of musical scale features. This also makes it possible to analyze data

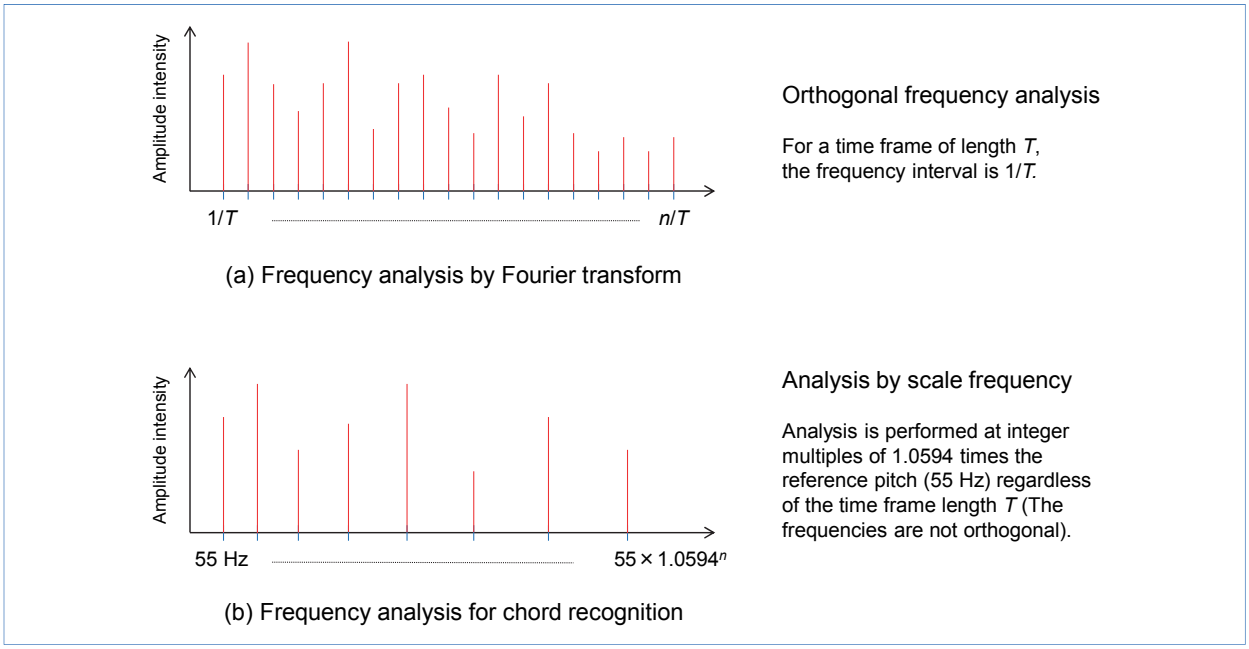

Figure 1 Frequency analysis method

*6 Base frequency: The discrete frequency unit used in frequency analysis.

with an arbitrary frame duration, so that fast melodies can be analyzed for a short period of time, and slow melodies for a longer period of time.

Since there are 12 musical scales if notes an octave apart are regarded as the same, it is also possible to reduce the computation time by using a 12-dimensional*7 chroma vector where the amplitude intensities of the same notes in different octaves are added together. To see the differences between octaves, it is possible to increase the number of dimensions of the chroma vector. If a four-octave musical scale is analyzed, then this results in a 48-dimensional chroma vector.

2.3 Using a Chroma Vector for Chord Recognition

Even with chroma vectors alone, it is possible to recognize chords in music. For simple triads (three-note chords), even considering just major and minor chords, there are two types of chord for each of the 12 musical scales, making a total of 24 types of chord. Since the constituent notes of each chord are fixed, we calculate the inner products of the chroma vector with binary vectors that contain 1 for notes that are included in a chord, and 0 for notes that are not included in the chord. Out of 24 different chords, the one that produces the largest value is output, thereby implementing a chord recognizer.

This method works well with notes produced by a single instrument. However, to recognize the chord progressions in tunes that combine multiple instruments with percussion and/or vocals, the recognition performance is not adequate. Therefore, in our technique, we apply a DNN to the chroma vectors so that musical chord progressions can be recognized more accurately.

3. Chord Recognition Technique

A chord recognition technique for music is able to analyze music data recorded on a CD or the like to produce sheet music showing the chords that are played. There are many different kinds of music, but in this article, we will concentrate on music played by bands that include drums, bass, guitar, keyboards, vocals and the like. In this technique, in order to accurately analyze the chords in this sort of music, we use a DNN to learn about the music data and chord progressions of existing music with the aim of improving accuracy. **Figure 2** shows the chord recognition procedure of this technique.

1) Input Music Data (Fig. 2 (1))

First, the music data is input into the chord recognizer. The music data is assumed to be in a format such as data recorded on a CD, and is input as a stereo sound source.

2) Beat*8 Detection (Fig. 2 (2))

To create the chord notation, the music must be divided into bars by detecting the beat based on the strength of the drums or instrument sounds. This is done by using the stereo music source directly, and estimating the beats from changes in the intensity of the music amplitude.

3) Vocal Cancellation (Fig. 2 (3))

With the aim of deleting vocals, percussion and the like to facilitate chord recognition, we cancel out the sound sources positioned at the center of the stereo sound source. The sound at the center can be canceled simply by adding the opposite phase of the right-channel signal to the left-channel signal. Since the sounds of instruments like guitars

*7 Dimension: The number of elements in the DNN input vector.

*8 Beat: A quarter-note period in music.

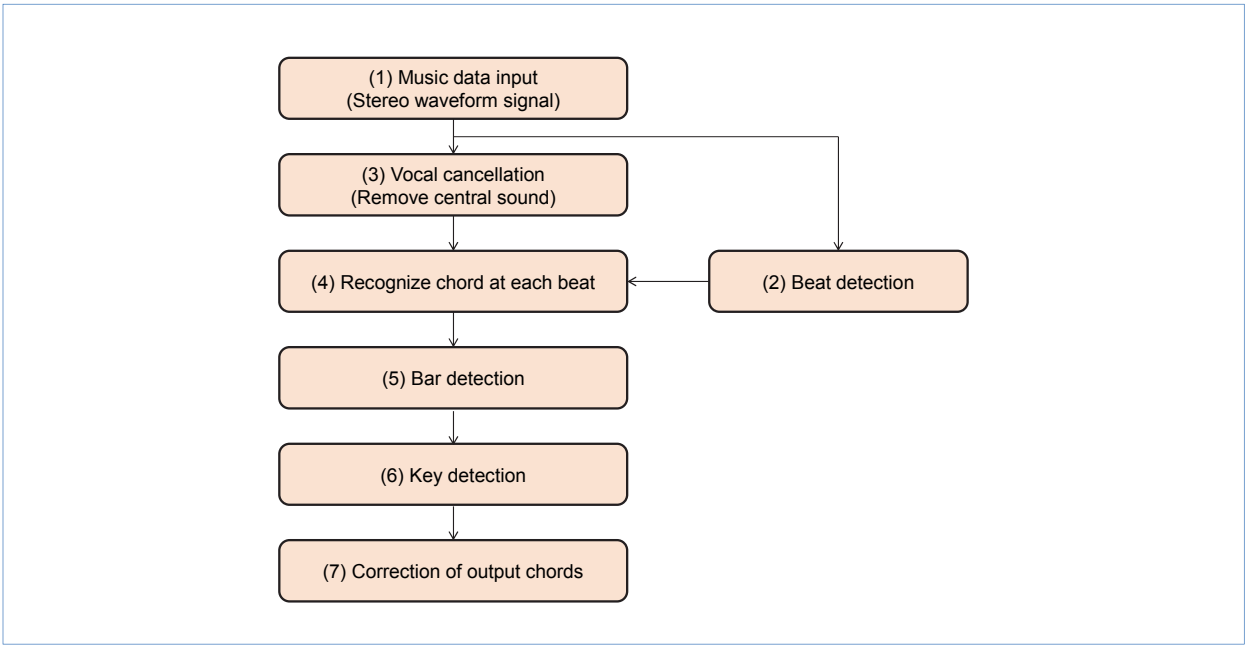

Figure 2 Procedure for analyzing chords in music

and keyboards are often positioned towards the left or right side, this operation leaves behind the accompanying music. Chord recognition is performed using sounds that are close to this accompaniment music.

4) Recognition of Chords for Each Beat (Fig. 2 (4))

In chord recognition, chord pattern recognition is performed from the chroma vectors as described above, but a DNN is used here. **Figure 3** shows the procedure for using a DNN to recognize chords from a song's waveform data.

(a) Generate a feature map for each beat

First, each detected beat is partitioned off as a time-domain[*9] segment. Chord recognition is applied to the segment from the timing of one beat until the timing of the next beat. Suppose the calculation of chroma vectors is performed in 80 ms units. If the duration of a single beat is 500 ms, then the chroma vectors will be calculated for approximately six frames per beat. If the chroma vector for a single frame is a 48-dimensional vector as a result of analyzing four octaves, i.e., if there are n frame regions in a single beat interval, then $48 \times n$ feature value maps can be generated.

(b) Generate 24-dimensional input vectors

This feature value map can be used as the input to the DNN, but since the time intervals for each detected beat are different, and since the feature value sizes are not fixed and may be redundant, we reduce the feature value map as in a Convolutional Neural Network (CNN)[*10] to produce a 24-dimensional fixed-length input vector. 24-dimensional vectors generated in this way are used as the input vectors of the DNN. This may be considered as a process corresponding to the convolutional layers[*11] and pooling layers[*12] of a CNN.

*9 Time domain: In signal analysis, this domain is used to show the temporal makeup of a signal's components. A time-domain signal can be converted to a frequency-domain signal by a Fourier transform.

*10 CNN: A neural network that performs pre-processing such as filtering on feature values to be input to a neural network and forms an input vector.

*11 Convolutional layer: A layer that emphasizes key features by applying a filter to the input feature value map.

*12 Pooling layer: A layer that compresses feature values by averaging or selecting redundant feature values with respect to the input feature values.

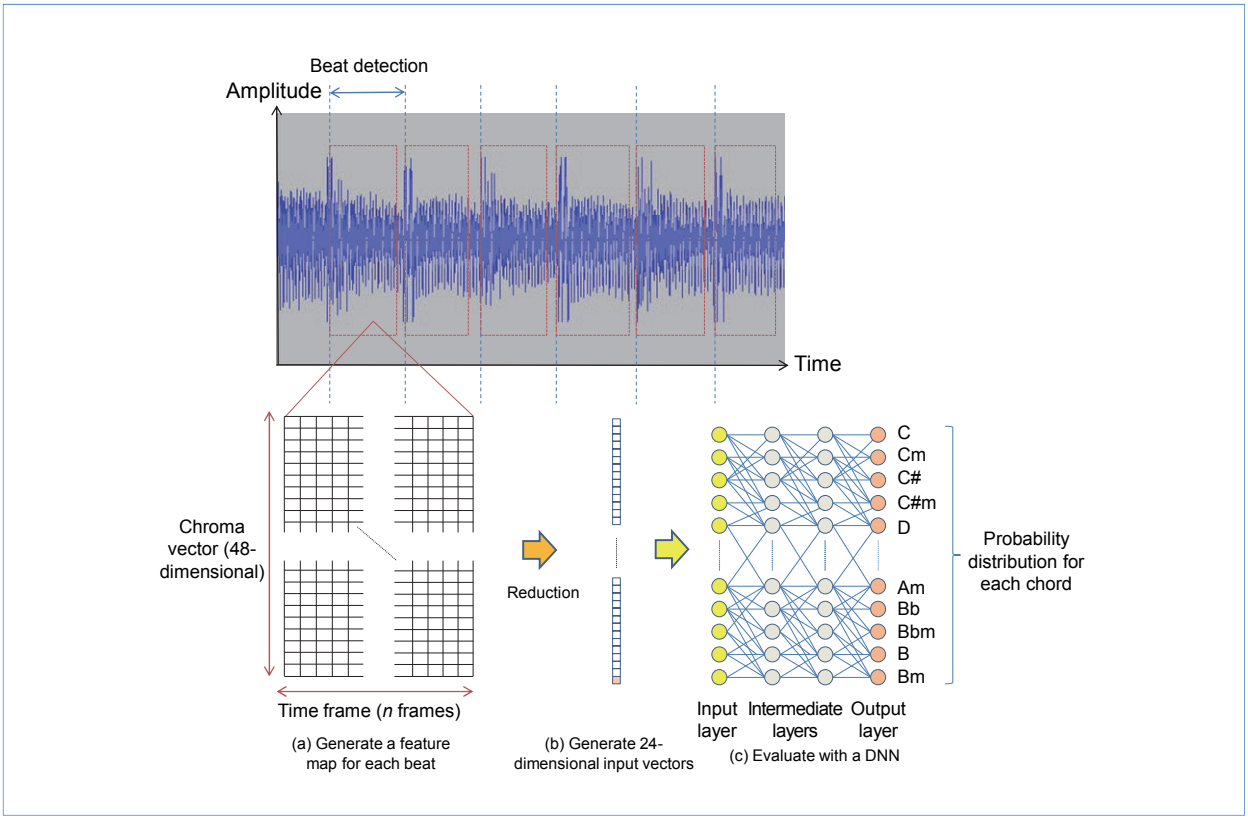

Figure 3 Chord recognition method

(c) Evaluate with a DNN

A DNN is formed with this input vector as the input layer values, and with two intermediate layers and an output layer that yields a probability distribution of 24 types with a softmax function*13. Although this technique can also be applied to tetrads (four-note chords) such as seventh*14 and diminished chords*15, we will concentrate on the evaluation of 24 different triads in this article in order to see the effect of the performance improvement.

For the learning data, we used CD music recordings saved as stereo WAVE files*16 with 15 bits per sample at a sampling rate of 44,100 Hz, and we performed learning using meta-files*17 recording the chords at each beat of the music.

Up to the point where we generated a feature value map for each beat and a reduced 24-dimensional input vector, the learning and evaluation processes are both the same. To train the DNN, we used music data consisting of 30 tunes played by a band. After training the network, we performed chord recognition using the music data of five tunes prepared separately from the training music data. For chord recognition, we compared the correct answer rates of recognition using the abovementioned method based on chroma vectors alone with that of the proposed method where a DNN is also used. The results of this comparison are shown in

*13 Softmax function: A function that normalizes a distribution function so that the sum total of all the output values becomes equal to 1.
*14 Seventh chord: A four-note chord obtained by adding the seventh note of the scale to the triad comprising the root, third and fifth notes.
*15 Diminished chord: A code obtained by adding the third and flattened fifth notes to the root note.
*16 WAVE file: A file format to store audio waveforms. The sampled audio data is stored without compression.
*17 Meta-file: In this article, a meta-file is a file containing data such as chord progressions and rhythm of musical content.

Figure 4. The correct answer rate depends on the tune data, but we found that the use of a DNN improves the correct answer rate by about 12% on average compared with chord recognition using only chroma vectors. We only used 30 tunes as learning data in this test, so it is possible that even greater correct answer rates could be achieved if a greater amount of learning data is used.

5) Bar[*18] Detection (Fig. 2 (5))

After recognizing the chords for each beat, the boundaries between bars are detected. This is done by using the fact that chord changes often take place between bars. Specifically, by lining up the chord recognition output results for each beat and assuming an ordinary rhythm of four beats to the bar, the first beat of the bar is recognized as the beat where there are the most chord changes.

6) Key Detection (Fig. 2 (6))

Next, the music's key is judged. This is needed in order to output the final chord recognition results correctly. The diatonic chords of the C major scale are shown at the top of **Figure 5**. Diatonic chords are chords that are often used in a particular key, so for example in the key of C major, these would be the set of triads formed by each note of the scale with the notes 3 and 5 steps further up (i.e., C-E-G, D-F-A, E-G-B, etc.). A tune in which this set of diatonic chords appears very frequently is highly likely to be in the key of C major. So if we calculate the rate of occurrence of diatonic chords for every scale (C major, C♯ major, D major, etc.), we can judge which key a tune is in by seeing which key's diatonic chords appear most often.

7) Correction of Output Chords (Fig. 2 (7))

Finally, the recognized chords are corrected. Even when chord recognition is performed using a DNN, only 70–80% of the chords are identified correctly. These incorrect chord recognition results can be somewhat improved by weighting the results based on the diatonic chord and secondary dominant chord to produce a final chord judgment.

The bottom part of Fig. 5 shows the diatonic chords for the key of D major. In the key signature of D major, the notes C and F are raised to

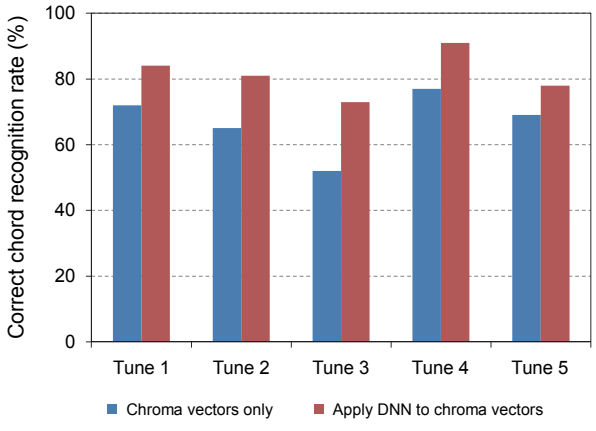

Figure 4 Correct chord recognition rate

*18 Bar: A unit time segment of a musical composition. In a composition written in common time, each bar consists of four beats.

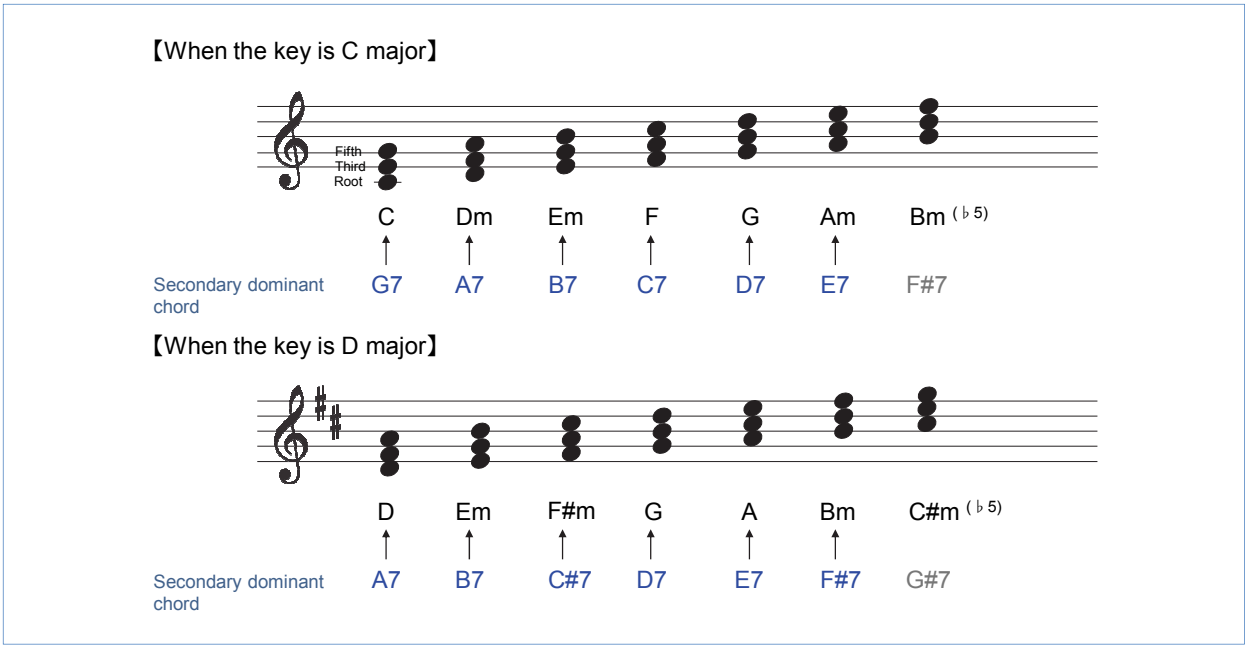

Figure 5 Diatonic chords and secondary dominant chords

C♯ and F♯, so this key uses a different set of chords than C major. There are many tunes that basically consist only of diatonic chords, but the next most frequently used chord is called the secondary dominant chord. This corresponds to a seventh chord based on the fifth note of each diatonic chord, as shown in blue in the figure. For the first chord on the left in Fig. 5, the chord on the fifth note is included in the normal diatonic seventh chord, so this is not regarded as a secondary dominant chord. Also, for the seventh diatonic chord (Bm), the root note[*19] (F♯7) of the seventh code on the fifth note cannot be used because it is outside the C major scale (C, D, E, F, G, A, B), so F♯7 is also excluded from the secondary dominant chords.

The chord recognition results are corrected by weighting these secondary dominant chords. The weighting calculations are performed by multiplying the probability distribution obtained from the DNN chord recognition results by the weightings of these chords, and taking the chords with the highest resulting values as the recognition results. This improves the chord recognition accuracy by about 5–10%. Based on the chord recognition results output by this process, it is possible to transcribe the music into chord notation. Although it is difficult to produce chord notation that matches the music exactly, it is possible to create chord notation that is useful for reference.

4. Ad-lib Performance Evaluation Function

In music, ad-lib refers to a playing style where the notes are played freely along with some accompaniment instead of playing notes exactly as they are written in a musical score. One way users can enjoy playing ad-lib is to play a free melody over

*19 Root note: The note at the base of the musical scale from which chords are formed. In most cases, it is the lowest note of the scale.

their favorite tune. Here, we will explain a performance evaluation function based on the above chord recognition technique that allows the user to perform ad-libs over a favorite tune.

First, when the user plays a favorite tune, the chords in this tune are recognized. At the same time, the bar boundaries and the key of the tune are also identified. While the music is playing, the performance evaluation system figures out which bar is currently being played, and which chords it contains. It then converts the user's performance into chroma vectors, and detects the pitch of the notes that the user is playing. If the pitch of the user's performance matches the constituent notes of the chords in the current bar, then the user's score is increased. If the user plays notes that are not included in these chords, a few points are still awarded if these notes are in the major scale of the tune's key. If the user plays a note that does not match either of these rules for more than one beat, then points are deducted. This scoring method provides a basic way of evaluating ad-lib performances.

In the ad-lib performance evaluation system implemented as an application of this technique, the user's ad-lib performance is analyzed while playing a tune, and the user is praised or cautioned if the score achieved in each bar is high or low. As a result, the user can experience the tension of being evaluated in real time while performing ad-lib.

5. Conclusion

In this article, we described a musical chord recognition technique that uses chroma vectors and a DNN. As an application example, we also introduced an ad-lib performance evaluation system. In the future, we hope to pursue more musical analysis functions and realize a musical agent.

REFERENCES

[1] M. Yamamoto, S. Naga and J. Shimizu: "Positive musical effects on two types of negative stressful conditions," Psychol Music, Vol.35, Issue 2, pp.249–275, Apr. 2007.

[2] H. Nakayama, F. Kikuta and H. Takeda: "A Pilot Study on Effectiveness of Music Therapy in Hospice in Japan," J Music Ther, NC46.2, pp.160–172, Jul. 2009.

[3] C. Harte and M. Sandler: "Automatic Chord Identification Using a Quantized Chromagram," in Proc. of Audio Eng. Soc., May 2005.