CHORD RECOGNITION USING PREWITT FILTER AND SELF-SIMILARITY

Nikolay Glazyrin
Ural Federal University
nglazyrin@gmail.com

Alexander Klepinin
Ural Federal University
alexandr.klepinin@usu.ru

ABSTRACT

In this paper we propose a method of audio chord estimation. It does not rely on any machine learning technique, but shows good recognition quality compared to other known algorithms. We calculate a beat-synchronized spectrogram with high time and frequency resolution. It is then processed with an analogue of Prewitt filter used for edge detection in image processing to suppress non-harmonic spectral components. The sequence of chroma vectors obtained from spectrogram is smoothed using self-similarity matrix before the actual chord recognition. Chord templates used for recognition are binary-like, but have the tonic and the 5th note accented. The method is evaluated on the 13 *Beatles* albums.

1. INTRODUCTION

Audio chord estimation is one of the most interesting tasks in music information retrieval. Representation of audio as a sequence of chords can be helpful for many other tasks, and itself can provide valuable information to the user. Chord recognition algorithms have been showing great progress during last years. 12 out of 18 algorithms that have participated the MIREX 2011 Audio Chord Description task have achieved chord weighted average overlap ratio greater than 0.70. All of them use machine learning algorithms to some extent, except one of the variants of the method by T. Cho and J. P. Bello [1] and the method proposed by T. Rocher et al. [2]. But the usage of machine learning makes the algorithm dependent on training data. Until recently the only data set available to researchers was composed of 12 albums of *The Beatles*, one 2-CD album of *Queen* and one album of *Zweieck*. Not a long time ago the Music Audio and Research Lab in New York University started to annotate the songs from RWC collection of popular music. Still, the available data cover only a small subset of world music. Therefore, we believe that the development of an algorithm that could provide good quality of chord recognition without any training makes sense.

The transition from low-level sound information recorded in the wave file to high-level information about the sequence of chords is usually done in several steps. At first the spectrogram of sound file is calculated. The Short-Time Fourier Transform and the constant-Q transform [3] are commonly used at this step. Then the spectrogram is transformed into a sequence of chroma vectors. Pitch Class Profile vectors, introduced by T. Fujishima [4], are often used here. Some modifications were proposed later, such as Enhanced Pitch Class Profile [5] and Harmonic Pitch Class Profile [6]. Chroma DCT-Reduced log Pitch (CRP) features introduced by M. Müller [7] also became popular recently.

The sequence of chroma vectors is then used to obtain the resulting sequence of chords. Two common methods used at this step are chord templates and probabilistic chord models [8]. The latter is more popular, but it usually requires a learning stage. The former requires no training and corresponding systems usually run faster than the systems based on probabilistic chord models. The system by L. Oudre et al. [9] uses 24 binary templates for major and minor chords and has low computational time. The system by T. Cho and J. P. Bello [1] also uses 24 chord templates in one of its variants. They employ the Viterbi algorithm to determine the most probable sequence of chords. The pseudo-probabilities for this algorithm variant are calculated by taking the reciprocal of the Euclidean distances. The quality of chord recognition shown by this method does not differ much from the quality of other participating algorithms (the variants of the same algorithm using gaussian mixture models; systems based on hidden Markov models). The system by T. Rocher et al. [2] tries to incorporate the musical theory into the process of chord sequence recognition. They estimate both chord and local key on each frame and then look for the best path through chord-key pairs.

The suggested method is a combination of well-known techniques and the new ideas. It differs from the methods listed above in the following aspects. We propose the new method of preliminary filtering of the spectrogram based on the analogue of Prewitt filter. The templates used for major and minor chord are not really binary. The templates for augmented and diminished chords are introduced as well, so the total number of templates used is 48. The sequence of chroma vector is smoothed using the self-similarity matrix.

This paper is organized as follows. Section 2 gives the description of suggested method. In section 3 the evaluation methodology is described. In section 4 the description of the results obtained by applying the method to the test corpus is given. We also analyze the influence of each step

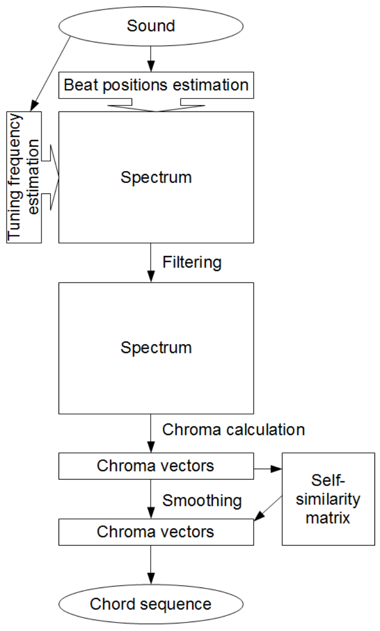

Figure 1. Block diagram of the proposed method.

on the results and typical errors of the proposed method. Section 5 summarizes the results and provides the plans for future work.

2. DESCRIPTION OF THE SYSTEM

The block diagram of the proposed method is shown at the figure 1.

2.1 Tuning

We used 16-bit 44100 Hz PCM wave files for the analysis. Stereo files were preliminary converted to mono. Before calculating the spectrum of the given audio recording the following steps have been performed.

2.1.1 Tuning frequency estimation

This step is important for recordings which were originally analogue. They may have variations or deviations in tempo, and the orchestra may be tuned to a frequency different from standard 440 Hz. For the majority of contemporary music this step is less important, because nowadays the music is often synthesized with the help of computers.

A very simple algorithm is used to estimate tuning frequency of an audio record. It is similar to the one that was described by Y. Zhu et. al. in [10]. This algorithm is by no means the best one, but it has been chosen for the ease of implementation.

The whole record is split into T sequential fragments. Then the constant-Q transform is applied on each fragment $t_i, i = 1, 2, ..., T$ with the following parameters: 120 frequency components per octave (or 10 components per note), minimal component frequency equal to 440 Hz, span of 4 octaves. Due to chosen (rather big) minimal frequency this procedure is very fast for the whole file. On each fragment t_i we determine the position of maximal spectrum value $g(t_i)$. Then a histogram of values of this function is constructed. It has 480 bins. The histogram is then folded to 10 bins: j-th bin of the original histogram contributes to $(j \ mod \ 10)$-th bin of the collapsed histogram for $j = 0, 1, ..., 479$. Then the position of maximal value in the resulting histogram shows the deviation of the tuning frequency from the base 440 Hz frequency in range from -1/2 to +1/2 semitone (427.5 Hz to 452.9 Hz) with step of 1/10 semitone (maximum at 0th position corresponds to no deviation). Calculated deviation is used further to specify the tuning frequency for main constant-Q transform.

2.1.2 Beat positions estimation

Musical events are often aligned according to musical meter. Therefore it makes sense to choose the fragments of sound file for further analysis aligned with musical beats. To estimate the position of each beat in the sound file, the *BeatRoot* library [11] was used.

The sequence of beat positions is then made T times more frequent by inserting $T - 1$ intermediate elements evenly between each pair of successive beat positions. Therefore, if the source sequence has n elements, then the resulting sequence will have $T \cdot (n - 1) + 1$ elements.

2.2 Calculation of the spectrogram

For each time position from this sequence the constant-Q transform on the fragment that is centered at this position is performed. The transform has the following parameters: 60 frequency components per octave, 4 octaves span, minimal component frequency is 33 semitones below tuning frequency. In case of standard tuning frequency 440 Hz the minimal component has frequency 65.41 Hz (corresponds to C2). The whole frequency range in this case spans from 65.41 Hz to 987.77 Hz.

The increase of the number of time positions is motivated by the fact that for high frequencies a very short fragment of the file is analyzed during the calculation of corresponding constant-Q transform components. Given the mentioned transform parameters and the sampling frequency of 44.1 kHz, the fragment of length 1.32 s is required to calculate the first transform component, and the length of 0.09 s is enough to calculate the last transform component. If a record has 120 beats per minute, a fragment between two successive beats will have length of 0.5 s. Thus it seems reasonable to perform more than one constant-Q transform per beat and later take the average of resulting spectra. This makes the method more robust. We have tried $T = 2, 4$ and 8. Greater values cause too large overlap between successive transforms.

Then we perform some transformations of the resulting spectrogram to reduce the noise and capture the areas that reflect the sound of melodic musical instruments.

First transformation is to apply the median filter with window size w_1 to each row of the spectrogram (each row corresponds to a component frequency of constant-Q transform). We also tried the moving average filter, but its usage led to worse result. Median filter reduces the noise without smoothing the spectrogram too much, as moving average filter does.

The objective of the second transformation is to suppress non-harmonic spectral components. Many systems employ preprocessing of this kind to clean the spectrogram, we only mention some of them. M. Mauch and S. Dixon in [12] subtract a so-called background spectrum from the spectrogram and discard negative values. D. FitzGerald in [13] uses median filter to separate a spectrogram into a harmonic part and a percussive part. N. Ono et. al. in [14] use maximum a posteriori estimation for this separation.

We propose the method of non-harmonic spectral components suppression based on the analogue of Prewitt operator usually used in image recognition algorithms to detect edges. We need to detect the horizontal lines on the spectrogram and zero all the other spectral components. Horizontal lines usually correspond to melodic sounds of musical instruments that last for some time and form chords. The operator that was used has the following matrix:

$$P = \begin{pmatrix} -1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

The dimensionality has been chosen to better match the horizontal lines given constant-Q transform frequency resolution of 5 components per note. For each point of the spectrogram the convolution of spectrogram area centered at this point and operator matrix is calculated. If the result is less than 0, corresponding spectrogram value is set to 0, and is preserved otherwise.

On the first picture at the figure 2 the spectrum of the first 20 seconds of "Yellow submarine" is shown. The first picture is obtained after the application of the 1st median filter. The second picture shows the spectrogram after the application of Prewitt filter.

Next, we compensate the initial increase of the number of time positions by shrinking the spectrogram in T times along the time axis. It is split into sequence of time fragments each consisting of T columns. For each fragment we calculate the arithmetical mean of its columns and replace the whole fragment with this new column. Therefore the spectrogram becomes beat aligned: each column corresponds to a beat position in the original file.

Finally, we apply one more median filter with window size w_2 to smooth the noise that could be added by this shrinking.

2.3 Calculation of chroma vectors

The resulting spectrogram is of size $240 \times n$. To obtain the sequence of 12-dimensional chroma vectors, we firstly fold the spectrogram from 4 octaves to 1 octave. It is done by summing the rows with indexes $j, j+60, j+120, j+180$ for each $j = 0, ..., 59$ into one row. This results in a matrix of size $60 \times n$ consisting of 60-dimensional pitch

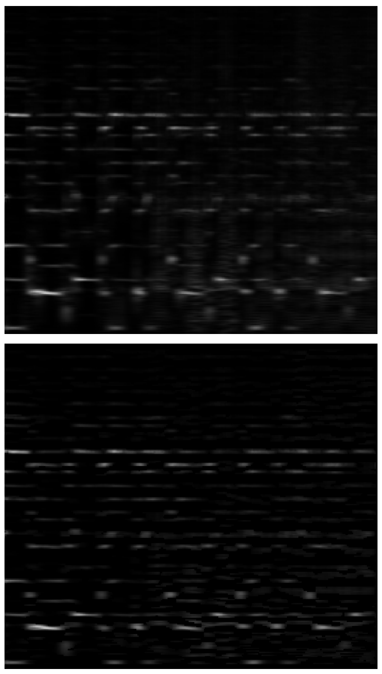

Figure 2. Spectrum before and after Prewitt filtering.

class profile column vectors. Then for each column vector c_i we calculate the 12-dimensional chroma vector p_i:

$$p_i[j] = \sum_{h=-2}^{2} c_i[5j-h] \cdot d^{|h|}, \quad i = 0, ..., n, \quad j = 0, ..., 11$$

The parameter d adjusts the contribution of spectral components that do not correspond to real notes. We choose $d = 0.6$. For the computation of $p_i[0]$ the components $c_i[58]$ and $c_i[59]$ are used.

2.4 Chroma vector sequence smoothing

The idea behind this process is that usually the same sequence of chords is repeated several times during one song. For example, this can be the same melody that sounds in 2 different verses or refrains. Suggested smoothing process is similar to the technique of recurrence plots used by T. Cho and J. P. Bello in [1]. But we do not group the chroma vectors into n-grams here.

We compute the self-similarity matrix $S = \{s_{ij}\}$ for the sequence of chroma vectors using Euclidean distance. This matrix has zeroes on the main diagonal. We normalize it so that $0 \leq s_{ij} \leq 1$ for any i, j. Then for each row we preserve only $M \cdot n$ minimal values and set all the others to 1 (here $0 \leq M \leq 1$). After that we only preserve those diagonal segments of that matrix which are parallel to main diagonal and have length greater than or equal to k: $\{s_{i+l,j+l} < 1 | 0 \leq l \leq k_1, k_1 \geq k\}$. The resulting matrix is "sparse" in the sense that it has few elements different from 1.

Then we recalculate the sequence of chroma vectors:

$$\hat{p}_i = \frac{\sum_{j=1}^{n} (1 - s_{ij}) \cdot p_j}{\sum_{j=1}^{n} (1 - s_{ij})}$$

The resulting sequence $\{\hat{p}_i\}_{i=1}^{n}$ is then used for chord estimation.

2.5 Chord estimation

We use chord templates to determine the corresponding chord for each chroma vector. We tried two well known metrics here: Euclidean distance and Kullback-Leibler divergence. The latter is not symmetric and can be calculated in 2 ways, both of them were tried. The chord corresponding to the template which is the nearest to current chroma vector is returned as a result.

The templates for major, minor, augmented, and diminished chords are used in the proposed method. A template is a 12-dimensional vector. For augmented and diminished chords this vector is binary. It has 1 on the positions corresponding to notes which compose the chord and 0 on other positions. But the templates for major and minor chords are not really binary. Template components that correspond to tonic and 5th note were additionally emphasized, so that the template for C:maj looks like (1.3, 0, 0, 0, 1, 0, 0, 1.3, 0, 0, 0, 0). Before distance calculation both template vector and chroma vector are normalized to have unit length in Euclidean space. We only mark as "no chord" the fragment that goes before the 1st beat detected by *Beatroot* in each audio record.

3. EVALUATION

The algorithm was evaluated on a collection of 13 Beatles albums annotated by C. Harte [15]. 4 songs have been excluded from the collection because of *Beatroot* beat detection errors on them: "A Day in the Life", "Strawberry Fields Forever", "Helter Skelter", "Revolution 9".

For each track the frame-based recall has been calculated (as it is defined in MIREX 2011 Audio Chord Estimation task) with the evaluations on the triad level. This means that only the chords from this list have been checked: N, X:maj, X:min, X:aug, X:dim, X:sus2, X:sus4. All the other chords (including all 7th chords) have been excluded from recall calculation. For those tracks that contain other chords the effective length (that was used for recall calculation) is therefore less than the whole track length. The details of the procedure can be found in [16]. Then the chord average overlap ratio (AOR) and chord weighted average overlap ratio (WAOR) were calculated for the whole collection using following formulae:

$$AOR = \frac{1}{C}\sum_{m=1}^{C} r_m, \quad WAOR = \frac{\sum_{m=1}^{C} l_m \cdot r_m}{\sum_{m=1}^{C} l_m}$$

Here C is the number of tracks in the collection, r_m and l_m are the frame-based recall and effective length for track m correspondingly. We employ these 2 overlap ratios as the measures of chord recognition quality.

	AOR	WAOR
Our method	0.7046	0.7125
Rocher et al. (RHRC1)	0.7289	0.7151
Cho and Bello (CB1)	0.7955	0.7786

Table 1. Average overlap scores

T	AOR	WAOR	Time
2	0.5525	0.5614	3285 s
4	0.6717	0.6813	5083 s
8	**0.7046**	**0.7125**	8675 s

Table 2. Impact of time positions increase factor T

4. RESULTS

The results of our method compared to the results of 2 systems participated in MIREX 2011 Audio Chord Estimation task are shown in table 1. These systems do not employ any machine learning algorithms and their recognition quality was evaluated using the same metrics. Therefore they are good candidates for comparison. The values were taken from MIREX 2011: Audio Chord Description result page [17].

We can see that the results of our method and system by Rocher et al. [2] are quite similar. But the system proposed by Cho and Bello [1] due to the usage of Viterbi decoder outperforms our method.

4.1 The impact of algorithm parameters

The proposed method depends on a number of parameters. The process of finding their best values may be considered as a "learning" or adaptation of the method to the collection of *The Beatles* songs. But we would rather investigate the influence of different parameters on a chord recognition quality. The parameter space is multidimensional, so we only investigate the impact of each parameter near the optimal point by fixing all parameters except analyzed one.

Here we provide their optimal values found during evaluation on the collection:

- Time positions increase factor $T = 8$. Table 2 summarizes the values of overlap scores and the processing times for different values of T. Bigger values lead to better result, but the processing time for the whole collection increases accordingly.

- First median filter window size $w_1 = 17$. It corresponds to 2 beats. Table 3 summarizes the impact of this parameter on the recognition quality.

- Second median filter window size $w_2 = 3$. It corresponds to 3 beats. The value of $w_2 = 5$ leads to both overlap ratios less than 0.7.

- The number of preserved elements in each row of self-similarity matrix $M = 0.15$. Table 4 summarizes the impact of this parameter on the recognition quality.

w_1	AOR	WAOR
9	0.7013	0.7091
13	0.7038	0.7113
17	**0.7046**	**0.7125**
21	0.7017	0.7107

Table 3. Impact of the first median filter size

M	AOR	WAOR
0.05	0.6941	0.6996
0.10	0.7042	0.7095
0.15	**0.7046**	**0.7125**
0.20	0.7013	0.7088
0.25	0.6952	0.7033

Table 4. Impact of the part of preserved elements of self-similarity matrix

- The minimal length of diagonal segment on self-similarity matrix $k = 3$. Table 5 summarizes the impact of this parameter on the recognition quality. The optimal values are different for 2 overlap ratios, so we choose that one that minimizes weighted average overlap ratio.

The total processing time of the whole collection depends mostly on the value of T. For $T = 8$ the average time of processing of one file is about 50 seconds on a test PC (Intel Core i3-2100 @3.10 GHz). The participants of MIREX 2008 Audio Chord Detection task had shown run time from 827 to 7411 seconds on the same dataset. In 2009 L. Oudre et al. [9] reported the whole collection processing time of their method of less than 800 seconds. So our straightforward implementation of spectrogram calculation really needs to be optimized for better performance.

4.2 The impact of different processing steps

Table 6 summarizes the impact of different processing steps of our method on the results. Our method without any additional processing was chosen as a baseline. Out of 2 variants of Kullback-Leibler divergence the best one was chosen — the distance from template to calculated chroma vector. But its advantage over the Euclidean distance is unnoticeable. The most-effective steps (except median filtering steps that reduce the noise considerably) are the application of Prewitt operator to the spectrogram and smoothing using self-similarity matrix.

The two most common types of errors of the proposed method are:

1. confusions between chords where root note of one chord is the 5th note for another chord (e.g. A:maj vs. D:maj);

2. recognition of a chord where no chord is playing;

3. confusions between major and minor chords with same root (e.g. E:maj vs. E:min).

k	AOR	WAOR
2	0.7001	0.7078
3	0.7046	**0.7125**
4	0.7053	0.7122
5	**0.7056**	0.7121
6	**0.7056**	0.7118
7	0.7026	0.7081

Table 5. Impact of the minimal length of diagonal segment of self-similarity matrix

Step	AOR	WAOR
Baseline	0.5445	0.5439
With tuning frequency estimation	0.5722	0.5725
With 1st median filter	0.6494	0.6509
With Prewitt operator	0.6604	0.6626
With 2nd median filter	0.6741	0.6779
Kullback-Leibler divergence	0.6751	0.6795
Non-binary major/minor templates	0.6801	0.6844
With self-similarity smoothing	0.7046	0.7125

Table 6. Impact of different steps

The errors of the first type may have been caused by the usage of templates with emphasized root and 5th notes. We have also tried non-emphasized, really binary templates with 1 to 6 additional harmonics, as in [9], but the recognition quality was not improved.

The errors of the second type are generally caused by the fact that we do not detect chord absence anywhere except the very beginning of the audio record. The regions of silence or where only percussive instruments are playing should be also detected.

The errors of the third type are very common for chord recognition systems. Our chord templates also do not assist in resolving this issue. We expect that proper local key estimation and the theory of harmony can be helpful here.

Another important problem is the blurring of borders between chords due to 2 steps of median filtering. To overcome this the additional segmentation step can be introduced. At that step we can locate exact borders between chords, and then retry the recognition considering those borders. Also this whole procedure can be made iterative.

5. CONCLUSION

The chord recognition method we presented shows good recognition quality compared to other known methods that do not utilize machine learning algorithms. The usage of Prewitt-like operator for horizontal lines separation on the spectrogram has positive effect on the recognition quality. The additional chroma vector sequence smoothing procedure also improves the result. Unlike many other methods we mark out the augmented and diminished chords. But because of the large number of constant-Q transforms the performance of the proposed method is far from perfect. In future we plan to optimize the algorithm to reduce the computation time. Also we plan to check the behavior of our method on a collection of popular music from RWC music database. We consider applying more musical knowledge to the process of chord extraction, especially the theory of harmony.

6. REFERENCES

[1] T. Cho and J. P. Bello, "A feature smoothing method for chord recognition using recurrence plots." in *ISMIR*, A. Klapuri and C. Leider, Eds. University of Miami, 2011, pp. 651–656. [Online]. Available: http://dblp.uni-trier.de/db/conf/ismir/ismir2011.html#ChoB11

[2] T. Rocher, M. Robine, P. Hanna, and L. Oudre, "Concurrent estimation of chords and keys from audio." in *ISMIR*, J. S. Downie and R. C. Veltkamp, Eds. International Society for Music Information Retrieval, 2010, pp. 141–146. [Online]. Available: http://dblp.uni-trier.de/db/conf/ismir/ismir2010.html#RocherRHO10

[3] J. Brown, "Calculation of a constant q spectral transform," vol. 89, no. 1, pp. 425–434, 1991.

[4] T. Fujishima, "Realtime chord recognition of musical sound: a system using common lisp music," *Proc. ICMC, 1999*, pp. 464–467, 1999. [Online]. Available: http://ci.nii.ac.jp/naid/10013545881/en/

[5] K. Lee, *Automatic chord recognition from audio using enhanced pitch class profile*. Citeseer, 2006, p. 306313. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.4283&rep=rep1&type=pdf

[6] E. Gómez, "Tonal description of music audio signals," Ph.D. dissertation, University Pompeu Fabra, Barcelona, Spain, July 2006. [Online]. Available: http://www.iua.upf.es/~egomez/thesis/

[7] M. Müller and S. Ewert, "Towards timbre-invariant audio features for harmony-based music," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 649–662, 2010.

[8] T. Cho, R. J. Weiss, and J. P. Bello, "Exploring Common Variations in State of the Art Chord Recognition Systems," in *Proc. Sound and Music Computing Conference (SMC)*, Barcelona, Spain, Jul. 2010, pp. 1–8.

[9] L. Oudre, Y. Grenier, and C. Févotte, "Chord recognition using measures of fit, chord templates and filtering methods," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New York, USA, 2009, pp. 9–12.

[10] Y. Zhu, M. S. Kankanhalli, and S. Gao, "Music key detection for musical audio," in *MMM*, 2005, pp. 30–37.

[11] "BeatRoot," http://www.eecs.qmul.ac.uk/~simond/beatroot/, 2012.

[12] M. Mauch and S. Dixon, "Approximate note transcription for the improved identification of difficult chords," in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, 2010.

[13] D. Fitzgerald, "Harmonic/percussive separation using median filtering," *Audio*, no. 1, pp. 10–13, 2010. [Online]. Available: http://arrow.dit.ie/argart/9/

[14] N. Ono, K. Miyamoto, H. Kameoka, and S. Sagayama, "A real-time equalizer of harmonic and percussive components in music signals." in *ISMIR*, J. P. Bello, E. Chew, and D. Turnbull, Eds., 2008, pp. 139–144. [Online]. Available: http://dblp.uni-trier.de/db/conf/ismir/ismir2008.html#OnoMKS08

[15] C. Harte, M. B. Sandler, S. A. Abdallah, and E. Gómez, "Symbolic representation of musical chords: A proposed syntax for text annotations," in *ISMIR*, 2005, pp. 66–71.

[16] "MIREX Audio Chord Estimation," http://www.music-ir.org/mirex/wiki/2011:Audio_Chord_Estimation, 2012.

[17] "MIREX Audio Chord Estimation results," http://nema.lis.illinois.edu/nema_out/mirex2011/results/ace/, 2012.