Musical Sound Recognition

Say Wei FOO and Chen Hwi NG
Department of Electrical Engineering
National University of Singapore
4 Engineering Drive 3, Singapore 117576

Abstract

In this paper, two different methods to implement one particular aspect of music transcription – musical note recognition – are presented. Both methods make use of existing algorithms in the form of the Constant-Q Transform (CQT) and the Discrete Wavelet Transform (DWT). Each existing algorithm is modified or extended so that a musical note recognition algorithm can be implemented on computer. Two main principles behind each method are peak detection using suitable thresholds and check for presence of harmonics in the process of note identification. The CQT-based method has a higher degree of accuracy, as it is able to resolve up to four simultaneous notes played on a guitar. The DWT-based method is less accurate but some suggestions are given as to how the algorithm can be further modified for improved performance.

1. Background

Transcription of music is defined as the act of listening to a piece of music and writing down music notation for the notes that constitute the piece [1]. Besides the area of note recognition, which is addressed in this paper, it also involves other areas like timing (rhythm tracking) and instrument recognition.

The transcription of *polyphonic* music (where several notes are played simultaneously), in particular, requires a lot of musical experience. This gives a motivation for research into the development of computer algorithms to *recognize* the contents of a musical signal and produce the musical score or some other form of symbolic representation

The CQT and DWT are chosen because they both provide variable frequency resolution in a way that can be made use of in note recognition algorithms. The CQT is, in fact, designed for the musical domain. The DWT, however, needs to be modified into a form that is more suitable for use in musical note identification.

2. CQT-Based Method

A. Theory of CQT

The CQT [2] uses windows of variable length so as to achieve variable frequency resolution. Variable resolution is desired because of the relation between frequencies of musical notes as shown in Eqn. (1).

$$f_{n+1} = 2^{1/12} f_n \qquad (1)$$

where f_{n+1} and f_n are the fundamental frequencies of the (n+1)th and nth notes respectively. This corresponds to an approximate 6% increase in frequency, known as *semitone spacing*. To resolve 6% differences in frequency, we need a variable resolution of about 3% of the frequency, or *quartertone* resolution. The ratio of frequency to resolution, Q, required is constant and equal to f / (0.03f) or 34.

With quartertone resolution, the frequency of the kth frequency component is

$$f_k = (2^{1/24})^k f_{min} \qquad (2)$$

where f_{min} would be the lowest frequency about which information is desired

The frequency resolution is calculated as the sampling rate divided by the window size, i.e.

$$df_k = S / N \qquad (3)$$

where df_k is the frequency resolution, S is the sampling frequency and N is the window size (in number of samples).

With a fixed sampling rate, the window size must be variable to achieve variable resolution. Hence using Eqn. (3), and the definition of Q as $f_k/\delta f_k$, we obtain

$$N[k] = S / df_k = (S / f_k)\, Q \qquad (4)$$

where $N[k]$ is the window size to be used in the evaluation of the kth spectral component.

Eqn. (5) is the main equation for the CQT.

$$X[k] = \sum_{n=0}^{N[k]-1} W[k,n]\, x[n]\, e^{-j2pQn/N[k]} \qquad (5)$$

where $X[k]$ is the kth spectral component of the STFT, $x[n]$ is the nth sample of the signal, $W[k,n]$ is the window function of variable length $N[k]$ and $2pQ/N[k]$ is the digital frequency.

A Hamming window as shown in Eqn. (6) is used.

$$W[k,n]=a+(1-a)cos(2pn/N[k]) \qquad (6)$$

where $\alpha = 25/46$ and $0 \le n \le N[k] - 1$.

B. Efficient Computation of CQT

An efficient algorithm [3] that makes use of a form of Parseval's equation can be used to perform the transform. The algorithm states that for any two discrete functions of time *x[n]* and *y[n],*

$$\sum_{n=0}^{N-1} x[n]y^*[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]Y^*[k] \qquad (7)$$

where *X[k]* and *Y[k]* are the DFTs of *x[n]* and *y[n]* respectively, and *Y*[k]* is the complex conjugate of *Y[k].* Eqn. (5) is shown here, with k expressed as k_{cq}.

$$X[k_{cq}] = \sum_{n=0}^{N[k_{cq}]-1} W[k_{cq},n]\, x[n]\, e^{-j2pQn/N[k_{cq}]} \qquad (8)$$

By letting $k^*[k_{cq},n] = W[k_{cq},n]\, e^{-j2pQk_{cq}n/N[k_{cq}]}$ we get, from Eqn. (7) and (8),

$$\begin{aligned} X^{cq}[k_{cq}] &= \sum_{n=0}^{N-1} x[n]\, k^*[k_{cq},n] \\ &= \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, K^*[k_{cq},k] \qquad (9) \end{aligned}$$

where $K[k_{cq}, k]$ is the DFT of $k[k_{cq},n]$.

The elements of $K[k_{cq},k]$ are called the *spectral kernels* while the elements of $k[k_{cq},n]$ are the *temporal kernels*. The spectral kernels are real since the temporal kernels are conjugate symmetric. The spectral kernels can be computed once and stored for all future uses, hence the efficiency.

In the spectral kernel calculations, a Hamming window is used. The windows and the exponential are placed symmetrically about the center of the time interval. Eqn. (10) shows the result.

$$\begin{aligned} K[k_{cq},k] &= \sum_{n=0}^{N-1} W[n-(\frac{N}{2}-\frac{N[k_{cq}]}{2}), k_{cq}] \\ &\times e^{j2pQ(n-N/2)/N[k_{cq}]}\, e^{-j2pkn/N} \qquad (10) \end{aligned}$$

The window is zero outside the interval [(N/2–N[k_{cq}]/2), (N/2+N[k_{cq}]/2)].

C. Recognition Algorithm

The output of the CQT (i.e. the spectrum) is processed as described in the following to identify the musical notes.

Spectral peaks are found by *magnitude comparison.* This involves comparing the magnitude of the spectral point with its neighbours. The next step is to decide which of the detected peaks are actually fundamental frequencies of notes. A *threshold* parameter is defined to reject peaks with values smaller than the threshold value.

Peaks that pass through the threshold test are given the harmonic test. A parameter called *harmonics requirement* is defined. It indicates the number of harmonic frequency peaks that must be present for the suspected fundamental peak to be identified as a note. Harmonic peaks are not subjected to the threshold test. The harmonics requirement used will have to depend on the instrument used. The threshold and harmonics tests constitute the main principles behind the recognition algorithm.

An additional check for continuity is done to minimize erroneous detection of notes that passed the first two tests. The rationale behind this check is that a note should last for at least two time intervals. Therefore, for every frame processed, a check is done with both the preceding and succeeding frames to validate the notes detected. If a detected note is present only in one frame, then it is most probably an error and should not be identified as a note.

It is found that spectral kernels are near zero over most of the spectrum. Therefore, the computational speed of the algorithm can be improved by using only those values that exceed a certain minimum value. Values that do not exceed the threshold are approximated as zero and thus do not contribute to the final result of the CQT. For this purpose, the error incurred is calculated for various values of this threshold by adding up the absolute values of all the kernels that fall below this threshold and dividing by the sum of the absolute values of all the kernels. The results are shown in Fig. 1.

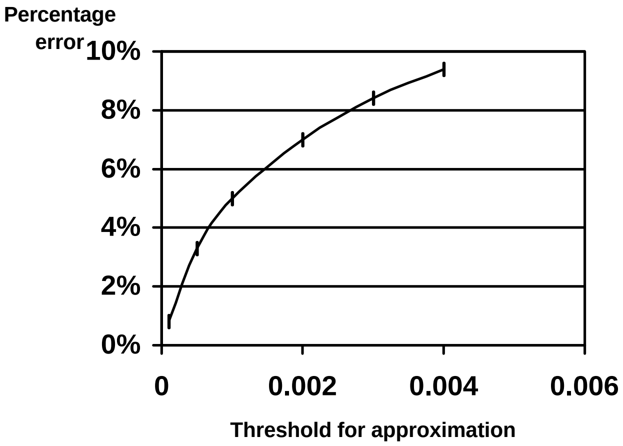

Fig. 1 Percentage error of kernel approximation for various threshold values

The threshold value of 0.001 is used as it gives a relatively low error percentage of 5.0% while providing a significant reduction of about 17.2 times in the number of kernel values used. With this approximation, the number of kernel multiplications in calculating the CQT is (16384 * 154/ 17.2 =) 146693. The number of complex multiplications in the calculation of the 16384 point FFT is (16384 $\log_2$ 16384 =) 229376. Thus, the total number of multiplications involved in this method is (146693 + 229376 =) 376069. The direct computation of CQT using Eqn. (5) involves FFTs of varying sizes for different frequencies and requires about 4109431 complex multiplications.

On the whole, the speed improvement of using the kernels and the above approximation is (4109431 / 376069 =) 10.9 times. It is also noteworthy that without any approximation, there is already a speed gain of about 1.5 times for the efficient calculation algorithm.

D. Performance of the Algorithm

Attention is focused on the performance of the algorithm on guitar recordings as it yields much better results than it does for piano recordings.

For our implementation, a 154-point CQT is used with the D2 note as the lowest identifiable note.

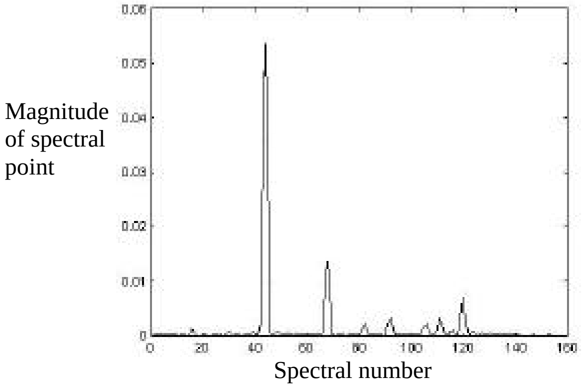

Fig. 2 CQT spectrum of a single guitar note (B3)

From Fig. 2, it is observed that there are 4 out of the first 5 harmonics present. The 4th harmonic at spectral number 99 is unnoticeable. The algorithm is able to identify B3 as the only note with a suitable threshold and a harmonics requirement of 4. This is because the note has a dominant fundamental and weak harmonics, thus a sufficiently high threshold will effectively eliminate the possibility of identifying any of the harmonics as a note.

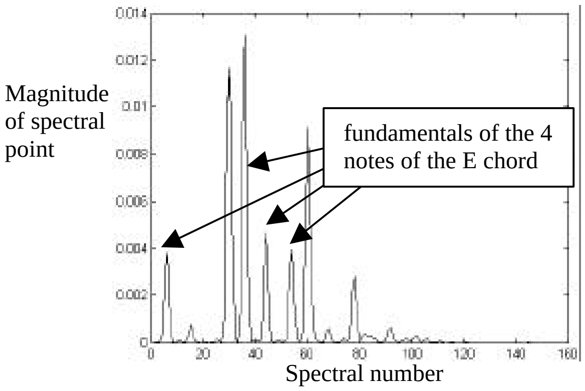

Fig. 3 4-note polyphony - CQT spectrum of a guitar E chord (E2, G3, B3, E4)

Frequency resolution for 4-note polyphony is shown in Fig. 3. The algorithm is able to identify the 4 notes present in the signal. However, the first harmonics of some of the notes are also mistakenly identified as notes. This is due to the prominence of the harmonic peaks, which are not rejected in the threshold test.

3. DWT-Based Method

A. Theory of DWT

Two common terms that are associated with the DWT [4][5] are *subband coding* and *multi-resolution analysis. Multi-resolution analysis* refers to the way the DWT analyzes signals using variable time and frequency resolution. The term, *subband coding* is used to describe the way the DWT decomposes the signal to achieve multi-resolution analysis.

The DWT employs two main procedures, which are applied repetitively to the signal. The first is *half-band filtering* and the second is *subsampling*. The signal is passed through a series of high-pass and low-pass *half-band filters* with *subsampling* at each stage, so as to analyze different frequency bands at different resolution.

A low-pass half-band filter removes frequency components that are above half of the highest frequency in the signal, hence the name half-band filter. Similarly, a high-pass half-band filter removes frequency components below half of the highest frequency. The highest frequency that is contained in a signal is equal to half of the sampling frequency, according to Nyquist's rule.

Subsampling refers to the reduction of the sampling rate of the signal. In the DWT, subsampling by two is done, meaning that every other sample is discarded, resulting in half the number of samples representing the same length of time (i.e. half the sampling rate).

The combination of half-band filtering followed by subsampling is the essence of multi-resolution analysis. After half-band filtering and subsampling, the time resolution is halved since only half the number of samples now characterizes the entire signal. However, the frequency resolution is doubled, since the frequency band of the signal now spans only half the previous frequency band, effectively reducing the uncertainty in the frequency by half.

The original signal is first passed through a low-pass half-band filter, h[n] and a high-pass half-band filter, g[n]. Each resulting sequence is subsampled by two, by discarding every other sample. Mathematically, this can be represented as shown in Eqn. (11) and (12).

$$y_{high}[n] = \sum_{n=-\infty}^{\infty} x[k]g[2n-k] \qquad (11)$$

$$y_{low}[n] = \sum_{n=-\infty}^{\infty} x[k]h[2n-k] \qquad (12)$$

where $y_{high}[n]$ and $y_{low}[n]$ are the outputs of the high-pass and low-pass filters, respectively, after subsampling by 2.

After the first round of decomposition (filtering and subsampling), the values of $y_{high}[n]$ are obtained as the Level 1 DWT coefficients. $y_{low}[n]$ is passed through a second round of the same decomposition procedure (using the same pair of half-band filters), to obtain another set of $y_{high}[n]$ and $y_{low}[n]$ values. This is continued successively as required or until two samples are left.

The decomposition process is illustrated in Fig.4.

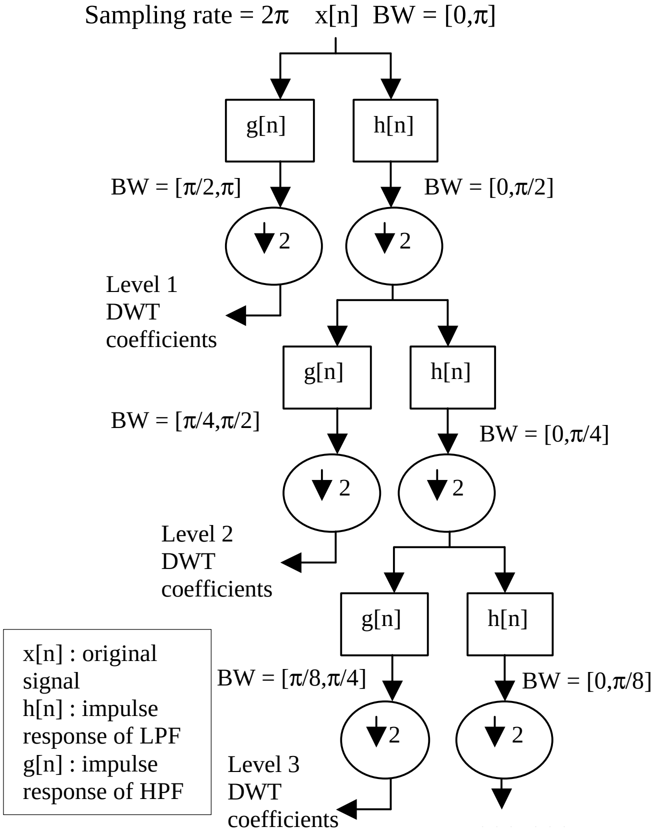

Fig. 4 Decomposition process of the DWT. The boxes denote half-band filtering while the circles represent subsampling by two.

The DWT of the original signal is obtained by concatenating all coefficients starting from the last level of decomposition. The DWT will then have the same number of coefficients as the original signal.

Frequencies that are most prominent in the original signal will appear as high amplitudes in that region of the DWT signal that includes those particular frequencies.

B. Recognition Algorithm

The frequency bands of the DWT are not designed for the musical domain, thus it is not readily usable for note recognition. The DWT is thus modified so that frequency bands that effectively separate out the musical frequencies are obtained.

The modified DWT is a direct extension of the conventional DWT, as it involves the same methods of decomposition (subband coding) described in Part A. Since the frequency bands of the DWT are not narrow enough, additional filtering has to be done to further split the frequency bands until each band contains at most one note frequency.

From Fig. 4, it can be seen that at each decomposition level, the DWT applies the two half-band filters only to the output of the low-pass filter of the previous level. Thus, to improve the frequency resolution of the high-pass bands, the decomposition is applied to the outputs of the high-pass filters as well. With reference to Fig. 4, this would result in a two-sided tree. However, in our method, the decomposition is not applied to *every* single high-pass or low-pass output. The decomposition is carried out until no frequency band contains more than one note frequency. Exhaustive filtering is not carried out since it is not necessary.

With the above modification, musical note recognition can then be conducted in the following way.

Time windows are applied to each frequency band containing a note frequency and the Mean Squared (MS) value of each time frame is calculated. Time windows are then translated at time intervals of 0.1 seconds to obtain the variation of MS values with time. The minimum translation time interval is restricted by the frequency bands at the lowest decomposition level, where a translation of one sample corresponds to a translation of 0.1 seconds. Window lengths are shown in Table 1, together with other information on the extraction of time frames.

Decomposition level	6	7	8	9	10	11
Time resolution (samples/sec)	344	172	86	43	21	10
Window length (in samples)	34	17	8	5	5	5
Translation of 0.1s (in samples)	34	17	8	4	2	1

Table 1 Window lengths and translations of the time window for each decomposition level containing useful frequency bands

Temporal 'peaks' are detected using the MS values of time frames. A temporal peak is an MS value that is greater than a predefined threshold, thus it indicates the presence of the particular frequency component, just like a spectral peak does in the CQT. Separate values are used for the *fundamental threshold* and the *harmonic threshold* since *fundamental peaks* are generally more prominent than harmonic peaks.

With the synchronization of the beginning of the time frames, spectral peaks are determined by examining the MS values of all the relevant frequency bands at a synchronized time frame. Just as before, different values are used for the fundamental threshold and the harmonic threshold. Temporal peaks are thus confirmed or rejected in the spectral peak detection process.

Each detected fundamental peak is then passed through the harmonics test just like in the CQT-based method. A fundamental peak will be identified as a note if there exist peaks at the frequency bands of its harmonic frequencies. The *harmonics requirement* is an adjustable parameter denoting the number of harmonics that must be present in order that a fundamental peak can be identified as a note.

C. Performance of the Algorithm

The next 3 figures show the variation of Mean Squared values with time for a single G3 note played on a guitar.

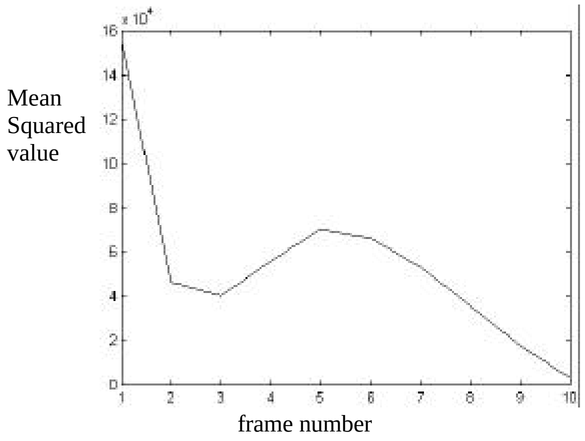

Fig. 5 Variation of MS values for the frequency band of the fundamental frequency of a single guitar note (G3)

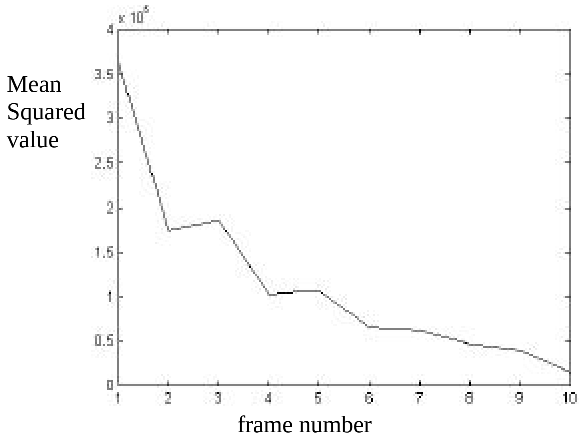

Fig. 6 Variation of MS values for the frequency band of the first harmonic frequency of a single guitar note (G3)

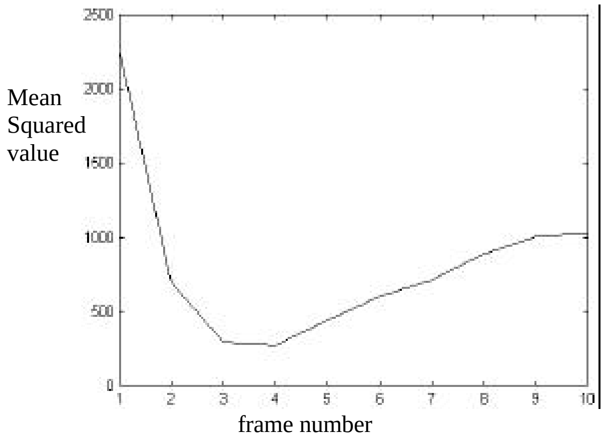

Fig. 7 Variation of MS values for the frequency band of the second harmonic frequency of a single guitar note (G3)

The MS values show a general decrease with time because of the fading of the sound. Observation of Figures 5, 6 and 7 shows that the fundamental and first harmonic have much larger MS values than the second harmonic. In fact, the first harmonic has larger values than the fundamental.

The algorithm is able to identify the G3 note when a check for the first 3 harmonics is made, using a fundamental threshold of 100 000 and a harmonic threshold of 1000. However, it also identifies the first harmonic as a note. Because the first harmonic has larger MS values than the fundamental, raising the fundamental threshold eliminated the fundamental and not the harmonic. It is also found that raising the harmonic threshold has the same effect as eliminating the fundamental instead of the first harmonic.

Accuracy decreases significantly with increasing polyphony. Low accuracy is partly due to insufficient frequency resolution resulting in significant spill-over into adjacent frequency bands. One possible improvement to the algorithm is to implement quartertone spacing by further dividing the spectrum (through half-band filtering) so that there is one buffer frequency band between the frequency bands of adjacent notes.

4. Conclusion

Two methods of musical note recognition are investigated in this paper. It is found that the CQT-based method gives higher accuracy than the DWT-based approach. The CQT-based method has the potential for further development to explore other aspects of music transcription like rhythm tracking and instrument recognition. Although the DWT-based method as investigated is less accurate, it may be considered as additional input for a music transcription system.

References

[1] Klapuri Anssi, "Automatic Transcription of Music", Tampere University of Technology (1998)

[2] Judith C. Brown, "Calculation of a constant Q spectral transform", J. Acoust. Soc. Am. Vol. 89 No. 1 (Jan 1991)

[3] Judith C. Brown, Miller S. Puckette, "An efficient algorithm for the calculation of a constant Q transform", J. Acoust Soc. Am. Vol. 92 No. 5 (Nov 1992)

[4] Albert Cohen, Robert D. Ryan, "Wavelets and Multi-scale Signal Processing", Chapman & Hall (1995)

[5] Mladen Victor Wickerhauser, "Adapted Wavelet Analysis from Theory to Software", A. K. Peters, Ltd (1994)