A MODIFIED CONSTANT-Q TRANSFORM FOR AUDIO SIGNALS

Cristiano N. dos Santos, Sergio L. Netto, Luiz W. P. Biscainho, and Danillo B. Graziosi

PEE/COPPE & DEL/POLI, UFRJ
Caixa Postal 68504, Rio de Janeiro, RJ, 21945-970, BRAZIL
{csantos,sergioln,wagner,danillo}@lps.ufrj.br

ABSTRACT

This paper proposes a new transform for musical audio signals. The new transform, so-called modified constant-Q fast filter bank (mCQFFB), yields a log-like description in the frequency domain with improved frequency response when compared to the standard DFT. The improved frequency response of the mCQFFB is achieved by interpreting the sliding-DFT (sDFT) as a multi-stage filter bank, and substituting the sDFT basic filter in this description by a higher-order filter with improved characteristics. The logarithm spectral scale of the mCQFFB is achieved by the adequate resampling of a chosen channel filter. We also discuss implementation issues related to the mCQFFB and exemplify its application to the analysis of musical signals.

I. INTRODUCTION

The discrete Fourier transform (DFT) is the standard tool to perform signal analysis in the frequency domain [1], when using a computer. However, the DFT is not completely suitable to analyze musical audio signals, as the resulting linear-frequency scale tends to concentrate too much information in the high-frequency region. In addition, in a filter-bank perspective, the DFT can be shown to present significant superposition between neighboring frequency bands. These two facts motivates the introduction of a new transform with a constant-Q (log-like) behavior in the frequency domain with improved frequency response when compared to the DFT. The new transform, hereafter called modified constant-Q fast filter bank (mCQFFB), is then proposed as a combination of two techniques previously known in the literature: the constant-Q transform (CQT), introduced in [2], [3], which yields a signal decomposition in a logarithmic frequency scale; and the fast filter bank (FFB), presented in [4], [6], which is characterized by low levels of interference between neighboring bands in the frequency domain.

To introduce the mCQFFB, this paper is organized as follows: In Sections II and III brief descriptions of the CQT and FFB are given. Section IV then presents the mCQFFB, combining the log-like frequency description of the CQT with the selective frequency response of the FFB. A computer experiment is included in Section V, illustrating the mCQFFB suitability to analyze musical audio signals.

II. CONSTANT-Q SPECTRAL TRANSFORMATION

Considering the usual musical notation, the semitone can be seen as the basic unit for measure of note intervals. The equal-temperament scale adopts a constant ratio of $2^{\frac{1}{12}}$ between the frequencies of notes located one semitone apart from each other, which means their difference is around 6%. As an example, along the audible spectrum, a single semitone down the superior edge (20000 Hz) spreads over 1123 Hz, while the same frequency range in Hz up the inferior edge (20 Hz) would be equivalent to 70 semitones (recall that a modern piano covers 88 notes)! This indicates that the linear-frequency description of the DFT gives an inefficient description for musical audio signals. For these signals, a log-like spectral analysis tends to be more appropriate, since it allows harmonic frequencies to be represented in equal intervals.

The standard CQT can analyze the signal into components given by the following frequencies:

$$f_k = \left(2^{1/12}\right)^{\alpha k} f_{\min}, \qquad (1)$$

for $k = 0, 1, \ldots, (N-1)$, where α defines the frequency resolution in fractions of a semitone. For instance, $\alpha = \frac{1}{2}$ yields a quarter-tone resolution, which corresponds to a selectivity factor Q equal to

$$Q = \frac{f_k}{(\delta f)_{\mathrm{CQT}}} = \frac{f_k}{\left(2^{1/24} - 1\right) f_k} \approx 34. \qquad (2)$$

In order to force a constant Q factor, one can then analyze the input signal using a distinct number of points to determine each frequency component, as given by

$$N_k = \frac{f_s}{(\delta f)_{\mathrm{CQT}}} = \frac{f_s Q}{f_k}. \qquad (3)$$

The index k associated to f_k in a corresponding DFT with N_k points and sample frequency f_s should be the selectivity factor Q itself. One can then modify the DFT, using a variable number of points as given in equation (3), to determine the CQT of a signal $x(n)$ as

$$X_{\mathrm{CQT}}(k) = \frac{1}{N_k} \sum_{n=0}^{N_k - 1} x(n) e^{-j\frac{2\pi}{N_k} Q n} \qquad (4)$$

0-7803-8484-9/04/$20.00 ©2004 IEEE

for $k = 0, 1, \ldots, (N-1)$.

In this framewok, to satisfy the Nyquist condition, we must have

$$\frac{2\pi Q}{N_k} < \pi \;\Rightarrow\; N_k > 2Q. \qquad (5)$$

It should be noticed that the CQT cannot be inverted, since the number of input samples is larger than the window size for higher frequencies.

III. FAST FILTER BANK

The sliding-DFT (sDFT) can be described in the z domain as

$$\mathrm{sDFT}(z) = \sum_{i=0}^{N-1} z^i W_N^{ki} = \frac{1 - \left(zW_N^k\right)^N}{1 - zW_N^k}. \qquad (6)$$

The FFB, to be reviewed below, is a modification of the filter-bank description of the sDFT as given in the following lemma.

Lemma A: For $N = 2^L$, the N-point sDFT(z) operator can be written in the following form:

$$\mathrm{sDFT}(z) = \prod_{i=0}^{L-1}\left[1 + \left(zW_N^k\right)^{2^i}\right]. \qquad (7)$$

Proof by finite induction: For $L = 1$, or $N = 2$, equation (6) becomes

$$\mathrm{sDFT}(z) = \frac{1 - \left(zW_2^k\right)^2}{1 - zW_2^k}, \qquad (8)$$

and equation (7) becomes

$$\mathrm{sDFT}(z) = 1 + zW_2^k, \qquad (9)$$

which are equivalent. Now assuming that equations (6) and (7) are equivalent for a given L, then

$$\frac{1 - \left(zW_{2^L}^k\right)^{2^L}}{1 - zW_{2^L}^k} = \prod_{i=0}^{L-1}\left[1 + \left(zW_{2^L}^k\right)^{2^i}\right]. \qquad (10)$$

Multiplying both sides by $1 + \left(zW_{2^L}^k\right)^{2^L}$, we get

$$\frac{1 - \left(zW_{2^L}^k\right)^{2^{L+1}}}{1 - zW_{2^L}^k} = \prod_{i=0}^{L}\left[1 + \left(zW_{2^L}^k\right)^{2^i}\right], \qquad (11)$$

which is the same as equation (10) for $L' = L + 1$. This completes the proof.

■

It should be noted that the computational complexity of the sDFT is in the order of N, as indicated in [5].

Example 1: From equation (6), channel 34 (associated to the quarter-tone CQT) of a 256-point sDFT is described by

$$\begin{aligned} H_{34}(z) &= \prod_{i=0}^{7}\left[1 + \left(zW_{256}^{34}\right)^{2^i}\right] \\ &= G_a^{0,34}(z)G_a^{1,68}(z)G_a^{2,136}(z)G_a^{3,16}(z) \\ &\quad G_a^{4,32}(z)G_a^{5,64}(z)G_a^{6,128}(z)G_a^{7,0}(z), \end{aligned} \qquad (12)$$

where $G_a^{i,j}(z) = (1 + z^{2^i} W_N^j)$, with $j = [(34 \times 2^i) \mod N]$. According to [4] and using the fact that $(34)_{10} = (00100010)_2$, a slightly different notation would be used, yielding

$$\begin{aligned} H_{34}(z) = {} & H_a^{7,34}(z)H_a^{6,17}(z)H_c^{5,8}(z)H_a^{4,4}(z) \\ & H_a^{3,2}(z)H_a^{2,1}(z)H_c^{1,0}(z)H_a^{0,0}(z), \end{aligned} \qquad (13)$$

where

$$\begin{cases} H_a^{i,j}(z) &= (1 + z^{2^{(B-1-i)}} W_N^{\tilde{j}}) \\ H_c^{i,j}(z) &= (1 - z^{2^{(B-1-i)}} W_N^{\tilde{j}}) \end{cases} \qquad (14)$$

with $\tilde{j}$ being the bit-reversed version of j in $(B-1)$ bits or, equivalently, $\tilde{j} = [(34 \times 2^{B-1-i}) \mod (N/2)]$. Thus, after using the property

$$-W_N^K = W_N^{\frac{N}{2}+K}, \qquad (15)$$

one can verify that equation (13) is identical to equation (12).

The magnitude response of channel 34 in the 256-point sDFT is depicted in detail by the dash-dotted line in Figure 1, where one can readily see the first sidelobes 13 dB below the channel passband.

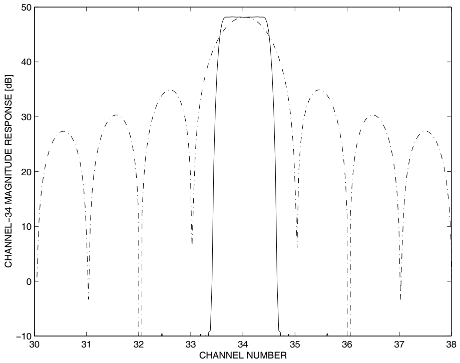

Fig. 1. Magnitude responses of channel 34 of the 128-point sDFT (dash-dotted line) and FFB (solid line).

■

The FFB is generated from the sDFT by substituting the first-order prototype filter in equation (7),

$$G_a^{0,0}(z) = 1 + z, \qquad (16)$$

by a set of halfband filters $G_a^{i,0}(z)$ (one for each level i) with improved frequency response [4]. Any FFB channel can then be determined in a similar fashion as in Example 1 above.

Example 2: The transfer function related to channel 34 in a 256-point FFB has a form similar to equation (12), with the corresponding $G_a^{i,0}(z)$ filters specified by the designer. The solid line in Figure 1 depicts the magnitude response of such channel using the filters given in Table I [4].

TABLE I

FFB SUBFILTER COEFFICIENTS [4] (ALL $g_a^{i,0}(0) = 1.0000$ AND $g_a^{i,0}(-n) = g_a^{i,0}(n)$).

i	$g_a^{i,0}(1)$	$g_a^{i,0}(3)$	$g_a^{i,0}(5)$	$g_a^{i,0}(7)$	$g_a^{i,0}(9)$	$g_a^{i,0}(11)$
0	0.62764	-0.18648	0.08816	-0.04299	0.01895	-0.00695
1	0.61750	-0.16013	0.05558	-0.01493		
2	0.57374	-0.07526				
3	0.56535	-0.06543				
4	0.50191					
5	0.50048					
6	0.50000					
7	0.50000					

■

IV. MODIFIED CONSTANT-Q FAST FILTER BANK

The combination of the constant-Q behavior of the CQT with the improved frequency response of the FFB yields a very interesting tool for the analysis of musical audio signals. However, such combination is not straightforward: The CQT performs the DFT of time-sequences with a variable number of samples, as given in equation (3), in order to control the channel bandwidths adequately. On the other hand, each FFB channel is built as a cascade of half-band filters, which would be analogous to require that the underlying DFT had been computed from a power-of-two number of samples. Therefore, to achieve the symbiotic combination between the CQT and the FFB techniques, we must overcome this discrepancy between the two techniques.

In [7], the arbitrary number of time-domain samples employed by the CQT was adapted to the power-of-two integer implicitly required by performing the resampling of the input signal. The resulting technique was the so-called constant-Q fast filter bank (CQFFB). In the present paper, however, we introduce the so-called modified constant-Q fast filter bank (mCQFFB), which is obtained by a generalization of the FFB technique via the resampling of the impulse response of one FFB channel filter. In that manner, the number of input-signal samples required by the DFT underlying to the modified FFB can be made equal to the number of samples required by the standard CQT. One should note that in the CQFFB the resampling operation is performed on-the-fly along the input signal, while in the mCQFFB the impulse responses of the filters (which are fixed) are resampled, meaning that this operation can be made only once, before the signal processing. This is the main direct advantage of the later over the former.

A proposed algorithm for the implementation of the mCQFFB is given in Table II. In this algorithm, f_s is the input-signal sampling frequency, $f_{\min}$ is the initial frequency to be analyzed, and α defines the spacing between two adjacent frequency samples in fractions of a semitone according to equation (1). See Example 3, below, for some possible values for these variables. In Step 2 of the proposed algorithm, the impulse response h_p of the prototype filter can be designed as an FFB channel filter. From equations (1) and (2), the selectivity factor is computed by

$$Q = \frac{1}{2^{\alpha/12} - 1}. \tag{17}$$

In Step 3, the resampling of the impulse response h_p can be performed by a general function

$$h_k = \texttt{resample}(h_{k-1}, f_k, f_{k-1}), \tag{18}$$

for $k = 2, \ldots, N$, with $h_1 = h_p$ and $f_1 = f_{\min}$.The resampling factor is given by the ratio between f_k (defined in equation (1)) and f_{k-1}.

TABLE II

ALGORITHM FOR IMPLEMENTING THE MCQFFB.

Step 1:	Define f_s, $f_{\min}$, and α.
Step 2:	Obtain the impulse response h_p of the prototype filter, with central frequency $f_{\min}$ and selectivity factor Q.
Step 3:	Resample h_p, changing its middle frequency to f_k in each filter.
Step 4:	Use the filters obtained in **Step 3** as the modified FFB channel filters.

V. COMPUTER EXPERIMENT

Example 3: To verify the usefulness of the mCQFFB technique, we generated a test signal $x(n)$ composed of three musical tones at 164.8138 Hz (E), 195.9977 Hz (G), and 261.6256 Hz (C). In this example, we implemented the mCQFFB algorithm with a sampling rate of $f_s = 44.1$ kHz; $f_{\min} = 130.8$ Hz, corresponding to C3, as the initial frequency sample; and $\alpha = 14$ (or Q=34), corresponding to a quarter-tone resolution, thus requiring a total of 24 channels per octave.

This signal was processed by the sDFT, FFB, CQT, and mCQFFB tools, all with 100 frequency bands between 130.8 and 2282.4 Hz (corresponding to C and halfway between C# and D, respectively), for the sake of uniformity. For the CQT and CQFFB, we had $Q = 34$, corresponding to a quarter-tone frequency resolution. Meanwhile, for the sDFT and FFB, which perform a linear frequency sampling, the resulting frequency resolution was equal to 21.5 Hz.

The output magnitudes obtained through all four analysis tools for the input signal $x(n)$ are depicted in Figure 2. In this figure, the peaks corresponding to a given tone (its fundamental frequency and related harmonics) are designated by the same symbol. In that way, characters '+', 'o', and 'x' indicate E, G, and C, respectively. From this figure, one can clearly notice the poorer low-frequency resolution yielded by the sDFT (Figure 2(a)) and the FFB (Figure 2(b)) due to their inherent linear frequency sampling: in particular, they did not identify the fundamental component of G, which was hidden by the fundamental component of E. In addition, by comparing the low-magnitude portions of each subfigure, one can observe the effects of lower selectivity in the magnitude responses of the sDFT (Figure 2(a)) and CQT (Figure 2(c)). Finally, Figure 2(d) depicts the mCQFFB magnitude response, which shows all three musical tones and their corresponding harmonics in a clear and sharp way.

■

VI. CONCLUSION

In this paper, we reviewed the constant-Q transform (CQT), presented in [2], and the fast filter bank (FFB), introduced in [4]. A novel transform, the so-called modified constant-Q fast filter bank (mCQFFB), was then proposed, by combining the CQT with a modified version of the FFB. The mCQFFB is then characterized by a log-like description with improved frequency response. These two features combined make the mCQFFB especially suited for audio applications, as illustrated by a computer experiment.

VII. REFERENCES

[1] P. S. R. Diniz, E. A. B. da Silva, and S. L. Netto, *Digital Signal Processing; System Analysis and Design*, Cambridge, UK, 2002.

[2] J. C. Brown, "Calculation of a constant Q spectral transform," *J. Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, Jan. 1991.

[3] J. C. Brown and M. S. Puckette, "An efficient algorithm for the calculation of a constant Q transform," *J. Acoustical Society of America*, vol. 92, no. 5, pp. 2698–2701, Nov. 1992.

[4] Y. C. Lim and B. Farhang-Boroujeny, "Fast filter bank (FFB)," *IEEE Trans. Circuits and Systems–II: Analog and Digital Signal Processing*, vol. 39, no. 5, pp. 316–318, May 1992.

[5] B. Farhang-Boroujeny and Y. C. Lim, "A comment on the computational complexity of sliding FFT," *IEEE Trans. Circuits and Systems–II: Analog and Digital Signal Processing*, vol. 39, no. 12, pp. 875–876, Dec. 1992.

[6] Y. C. Lim and B. Farhang-Boroujeny, "Analysis and optimum design of the FFB," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. II.509–II.512, London, UK, 1994.

[7] D. B. Graziosi, C. N. dos Santos, S. L. Netto, and L. W. P. Biscainho, "A constant-Q spectral transformation with improved frequency response," submitted to *IEEE Int. Symp. Circuits and Systems*, Vancouver, Canada, May 2004.

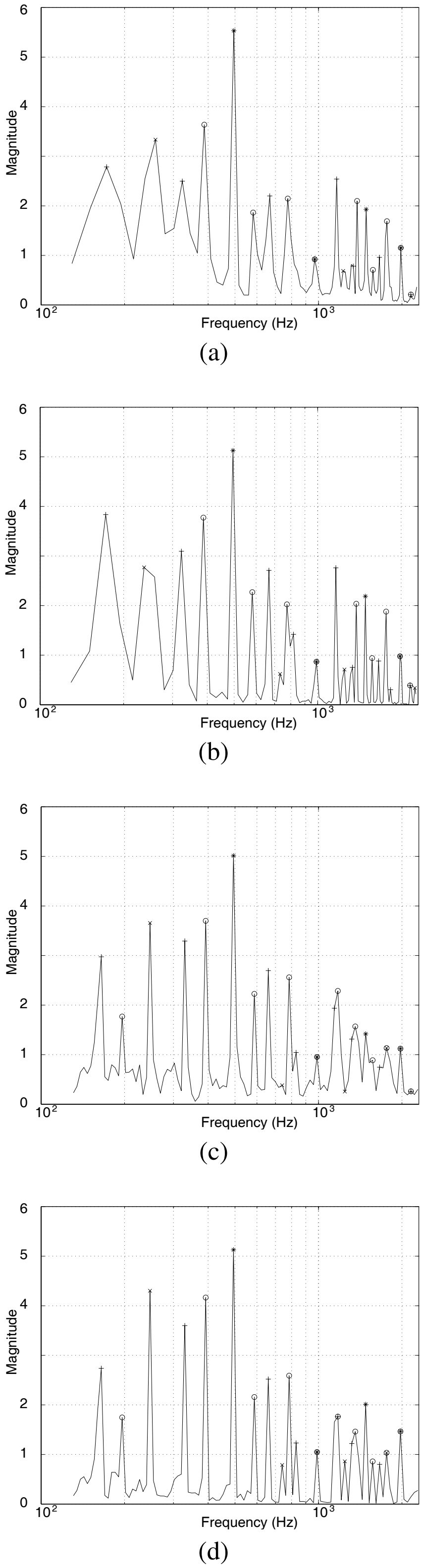

Fig. 2. Example 3: Magnitude responses of 100-point transformations of an input signal composed by three tones: (a) sDFT; (b) FFB; (c) CQT; (d) mCQFFB.