AN EFFICIENT MULTI-RESOLUTION SPECTRAL TRANSFORM FOR MUSIC ANALYSIS

Pablo Cancela **Martín Rocamora** **Ernesto López**
Universidad de la República, Instituto de Ingeniería Eléctrica, Montevideo, Uruguay
{pcancela,rocamora,elopez}@fing.edu.uy

ABSTRACT

In this paper we focus on multi-resolution spectral analysis algorithms for music signals based on the FFT. Two previously devised efficient algorithms (efficient constant-Q transform [1] and multiresolution FFT [2]) are reviewed and compared with a new proposal based on the IIR filtering of the FFT. Apart from its simplicity, the proposed method shows to be a good compromise between design flexibility and reduced computational effort. Additionally, it was used as a part of an effective melody extraction algorithm.

1. INTRODUCTION

Many automatic music analysis algorithms, such as those intended for melody extraction or multiple pitch estimation, rely on a spectral representation of the audio signal, typically the discrete Short Time Fourier Transform (STFT). A key issue that arises is the compromise between time and frequency resolution. The frequency components of a Discrete Fourier Transform (DFT) are equally spaced and have a constant resolution. However, in polyphonic music a higher frequency resolution is needed in the low and mid frequencies where there is a higher density of harmonics. On the other hand, frequency modulation gets stronger as the number of harmonic is increased, requiring shorter windows for improved time resolution. Thus, a multi resolution spectral representation is highly desired for the analysis of music signals. In addition, computational cost is a critical issue in real time or demanding applications so efficient algorithms are often needed.

In this context several proposals have been made to circumvent the conventional linear frequency and constant resolution of the DFT. The constant-Q transform (CQT) [3] is based on a direct evaluation of the DFT but the channel bandwidth Δf_k varies proportionally to its center frequency f_k, in order to keep constant its quality factor $Q = f_k/\Delta f_k$ (as in Wavelets). Center frequencies are distributed geometrically, to follow the equal tempered scale used in Western music, in such a way that there are two frequency components for each musical note (although higher values of Q provide a resolution beyond the semitone). Direct evaluation of the CQT is very time consuming, but fortunately an approximation can be computed efficiently taking advantage of the Fast Fourier Transform (FFT) [1].

Various approximations to a constant-Q spectral representation have also been proposed. The bounded-Q transform (BQT) [4] combines the FFT with a multirate filterbank. Octaves are distributed geometrically, but within each octave, channels are equally spaced, hence the log representation is approximated but with a different number of channels per octave. Note that the quartertone frequency distribution, in spite of being in accordance with Western tuning, can be too scattered if instruments are not perfectly tuned, exhibit inharmonicity or are able to vary their pitch continuously (e.g. glissando or vibrato). Recently a new version of the BQT with improved channel selectivity was proposed in [5] by applying the FFT structure but with longer kernel filters, a technique called Fast Filter Bank. An approach similar to the BQT is followed in [6] as a front-end to detect melody and bass line in real recordings. Also in the context of extracting the melody of polyphonic audio, different time-frequency resolutions are obtained in [2] by calculating the FFT with different window lengths. This is implemented by a very efficient algorithm, named the Multi-Resolution FFT (MR FFT), that combines elementary transforms into a hierarchical scheme.

In this paper we focus on multi-resolution spectral analysis algorithms for music signals based on the FFT. Two previously devised efficient algorithms that exhibit different characteristics are reviewed, namely, the efficient CQT [1] and the MR FFT [2]. The former is more flexible regarding Q design criteria and frequency channel distribution while the latter is more efficient at the expense of design constrains. These algorithms are compared with a new proposal based on the Infinite Impulse Response (IIR) filtering of the FFT (IIR CQT), that in addition to its simplicity shows to be a good compromise between design flexibility and reduced computational effort.

2. FIR Q TRANSFORM IMPLEMENTATIONS

2.1 Efficient constant Q transform

As stated in [3] a CQT can be calculated straightforwardly based on the evaluation of the DFT for the desired compo-

nents. Consider the kth spectral component of the DFT:

$$X[k] = \sum_{n=0}^{N-1} w[n]x[n]e^{-j2\pi kn/N}$$

where $w[n]$ is the temporal window function and $x[n]$ is the discrete time signal. In this case the quality factor for a certain frequency f_k equals k, since $Q_k = f_k/\Delta f = f_k N/f_s = k$. This corresponds to the number of periods in the time frame for that frequency. The digital frequency is $2\pi k/N$ and the period in samples is N/k. In the CQT the length of the window function varies inversely with frequency (but the shape remains the same), so that N becomes $N[k]$ and $w[n]$ becomes $w[n,k]$. For a given frequency f_k, $N[k] = f_s/\Delta f_k = f_s Q_k/f_k$. The digital frequency of the kth component is then given by $2\pi Q/N[k]$, the period in samples is $N[k]/Q$ and always Q cycles for each frequency are analyzed. The expression for the kth spectral component of the CQT is then [1] ,

$$X^{cq}[k] = \frac{1}{N[k]} \sum_{n=0}^{N[k]-1} w[n,k]x[n]e^{-j2\pi Qn/N[k]}. \quad (1)$$

Direct evaluation of equation (1) is time consuming, so an efficient algorithm for its computation has been proposed in [1]. The CQT can be expressed as a matrix multiplication, $X^{cq} = x \cdot T^*$, where x is the signal row vector of length N ($N \geq N[k]\ \forall k$) and T^* is the complex conjugate of the temporal kernel matrix T whose elements $T[n,k]$ are,

$$T[n,k] = \begin{cases} \frac{1}{N[k]} w[n,k]e^{-j2\pi Qn/N[k]} & \text{if } n < N[k] \\ 0 & \text{otherwise} \end{cases}$$

Computational effort can be improved if the matrix multiplication is carried out in the spectral domain. Using Parseval's relation for the DFT, the CQT can be expressed as,

$$X^{cq}[k] = \sum_{n=0}^{N-1} x[n]T^*[n,k] = \frac{1}{N}\sum_{k'=0}^{N-1} X[k']K^*[k',k] \quad (2)$$

where $X[k']$ and $K[k',\cdot]$ are the DFT of $x[n]$ and $T[n,\cdot]$ respectively. Spectral kernels are computed only once taking full advantage of the FFT. In the case of conjugate symmetric temporal kernels, the spectral kernels are real and near zero over most of the spectrum. For this reason, if only the spectral kernel values greater than a certain threshold are retained, there are few products involved in the evaluation of the CQT (almost negligible compared to the computation of the FFT of $x[n]$).

It is important to notice that although the original derivation of the CQT implies a geometrical distribution of frequency bins, it can be formulated using other spacing, for instance a constant separation. In the following, linear spacing is used to put all the compared algorithms under an unified framework.

[1] A normalization factor $1/N[k]$ must be introduced since the number of terms varies with k.

2.2 Multi-resolution FFT

A simple way to obtain multiple time-frequency resolutions is through the explicit calculation of the DFT using different frame lengths. In [2], an efficient technique is proposed where the DFT using several frame lengths is computed by means of the combination of the DFT of small number of samples, called elementary transforms. The idea arises from the observation that a transform of frame length N can be split into partial sums of L terms (assuming $N/L \in \mathbb{N}$),

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-\frac{j2\pi kn}{N}} = \sum_{c=0}^{\frac{N}{L}-1} \sum_{n=cL}^{(c+1)L-1} x[n]e^{-\frac{j2\pi kn}{N}}. \quad (3)$$

Each inner sum in equation 3 corresponds to the DFT of length N of a sequence $x_c[n]$, where $x_c[n]$ is an L samples chunk of $x[n]$, time-shifted and zero padded,

$$x_c[n] = \begin{cases} x[n], & cL \leq n < (c+1)L \\ 0, & \text{otherwise.} \end{cases}$$

So, it is possible to obtain a DFT of a frame of size N from N/L elementary transforms of frame size L, defined as

$$X_l[k] = \sum_{n=0}^{L-1} x[n+lL]e^{-j2\pi kn/N},\ l = 0, ..., \frac{N}{L} - 1.$$

To that end, it is enough to add the elementary transforms modified with a linear phase shift to include the time shift of $x_c[n]$, as stated by the shifting theorem of the DFT,

$$X[k] = \sum_{l=0}^{\frac{N}{L}-1} X_l[k]e^{-j2\pi kl/N}. \quad (4)$$

This procedure can be generalized to compute the DFT of any frame of length $M = rL$ by adding r elementary transforms ($r = 1, ..., N/L$) in the equation 4, which results in N/L possibles spectral representations with frequency resolutions of $f_s/(rL)$.

The computation of the multi-resolution spectrum from a combination of elementary transforms requires the windowing process to be done by means of convolution product in the frequency domain. Temporal windows of the form

$$w[n] = \sum_{m=0}^{\frac{M}{2}} (-1)^m a_m \cos\left(\frac{2\pi}{M} mn\right) \quad (5)$$

are suitable for this purpose because its spectrum has only few non-zero samples. Due to the fact that windowing is applied over zero-padded transforms, it is convenient to consider a periodic time window of the same length of the DFT to avoid the appearance of new non-zero samples of the window spectrum. In this case, the spectrum of a window of the form of equation 5 results in

$$W[k] = \sum_{m=0}^{\frac{M}{2}} (-1)^m \frac{a_m}{2} \left(\delta\left[k - m\frac{N}{M}\right] + \delta\left[k + m\frac{N}{M}\right]\right)$$

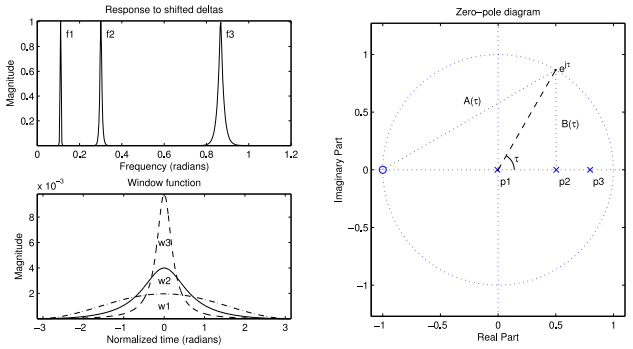

Figure 1. Zero-Pole diagram and IIR filters responses for three different input sinusoids of frequencies $f_1 = 0.11$, $f_2 = 0.30$ and $f_3 = 0.86$ radians.

For example, in Hann and Hamming windows only a_0 and a_1 are not zero and so its DFT contains solely three non-zero samples. As a counterpart, the restriction that $N/M = N/(rL) \in \mathbb{N}$ must be imposed, reducing the possible number of resolutions to $\log_2(N/L) + 1$.

3. IIR Q TRANSFORM

3.1 FIR/IIR Filterbank

The proposed methods define a Finite Impulse Response (FIR) filterbank with different impulse responses for different frequencies. The result of applying one of these filters can be regarded as multiplying the frame with a time window, which defines the time/frequency resolution. Variable windowing in time can also be achieved applying an IIR filterbank in the frequency domain. Let us define the k^{th} filter as a first order IIR filter with a pole p_k, and a zero z_k, as,

$$Y_k[n] = X[n] - z_k X[n-1] + p_k Y_k[n-1] \qquad (6)$$

Its Z transform is given by,

$$H_{f_k}(z) = \frac{z - z_k}{z - p_k}.$$

Here, $H_{f_k}(z)$ evaluated in the unit circle $z = e^{j\tau}$ represents its time response, with $\tau \in (-\pi, \pi]$ being the normalized time within the frame. A different time window for each frequency bin is obtained by selecting the value of the k^{th} bin as the output of the k^{th} filter.

The design of these filters involves finding the zero and pole for each k such that $w_k(\tau) = |H_{f_k}(e^{j\tau})|$, where $\tau \in (-\pi, \pi]$ and $w_k(\tau)$ is the desired window for the bin k. When a frame is analyzed, it is desirable to avoid discontinuities at its ends. This can be achieved by placing the zero in $\tau = \pi$, that is $z_k = -1$. If we are interested in a symmetric window, $w_k(\tau) = w_k(-\tau)$, the pole must be real. Considering a causal realization of the filter, p_k must be inside the unit circle to assure stability, thus $p_k \in (-1, 1)$. Figure 1 shows the frequency and time responses for the poles depicted in the zero-pole diagram.

This IIR filtering in frequency will also distort the phase, so a forward-backward filtering should be used to obtain a zero-phase filter response. Then, the set of possible windows that can be represented with these values of p_k is,

$$w_k(\tau) = \frac{(1-p_k)^2}{4}\left[\frac{A(\tau)}{B(\tau)}\right]^2 = \frac{(1-p_k)^2(1+\cos\tau)}{2(1+p_k^2-2p_k\cos\tau)} \qquad (7)$$

where $A(\tau)$ and $B(\tau)$ are the distances to the zero and the pole, as shown in Figure 1, and $g_k = (1-p_k)^2/4$ is a normalization factor[2] to have 0 dB gain at time $\tau = 0$, that is, $w_k(0) = 1$.

While this filter is linear and time invariant (in fact frequency invariant[3]) a different time window is desired for each frequency component. Computing the response of the whole bank of filters for the entire spectrum sequence and then choosing the response for only one bin is computationally inefficient. For this reason, a Linear Time Variant (LTV) system, that consists in a Time Varying (TV) IIR filter, is proposed as a way to approximate the filterbank response at the frequency bins of interest. It will no longer be possible to define the filter impulse response, as this could only be done if the filters were invariant to frequency shifts.

3.2 LTV IIR System

Selecting a different filter response of the filterbank for each frequency bin can be considered as applying an LTV system to the DFT of a frame. The desired response of the LTV for a given frequency bin is the impulse response of the correspondent filter.

Any LTV system can be expressed in the matrix form, $Y = K.X$ where K is the linear transformation matrix (also referred as Greens matrix) and, in this case, X is the DFT of the signal frame. A straightforward way to construct K for any LTV system is to set its i^{th} column as the response to a shifted delta $\delta[n-i]$, which is named Steady State Response (SSR).

The approach followed in this work consists in approximating the LTV system by a single TV IIR filter, assuming that the LTV system has a slow time varying behavior and that its SSR can be implemented by an IIR filterbank. Then it is verified that the approximation is sufficiently good for our purposes. In the case of variable windowing to obtain a constant Q, these assumptions hold, as time windows for two consecutive frequency bins are intended to be very similar, and the LTV system can be implemented by an IIR filterbank as seen before.

A direct way of approximating the IIR filterbank is by a first order IIR of the form of equation 6, but in which the pole varies with frequency ($p = p[n]$),

$$Y[n] = X[n] + X[n-1] + p[n]Y[n-1]. \qquad (8)$$

With an appropriate design, it reasonably matches the desired LTV IIR filterbank response, and its implementation has low computational complexity.

[2] This normalization factor can be calculated from the impulse response evaluated at $n = 0$, or by the integral of the time window function.

[3] Note that we will use the usual time domain filtering terminology in spite of the fact that filtering is performed in the frequency domain.

3.3 Time Varying IIR filter design

A question that arises is how to design the TV IIR filter in order to have a close response to that of the LTV IIR filterbank. Several design criteria have been proposed in the literature [7], that may depend on the problem itself.

The TV IIR can also be represented by a matrix K_v in a similar way as the LTV filterbank, so the design can be done as in [7], by minimizing the normalized mean square error, $E = ||K - K_v||_2/||K||_2$. In this work, the adopted design criteria is to impose the windows behavior in time in order to obtain the desired constant Q. Then, the error is regarded as the difference between the desired Q and the effective obtained value. It becomes necessary to define an objective measure of Q. Usually the quality factor of a passband filter is defined as the ratio between the center frequency and the bandwidth at 3 dB gain drop. In our case the filtering is done in the frequency domain, so it is reasonable to measure Q in the time domain. Given that Q represents the number of cycles of an analyzed frequency component in the frame, it makes sense to define Q as the number of cycles within the window width at a certain gain drop, for example 3 dB. If τ'_k is the time at this drop for frequency f_k, $w_k(\tau'_k) = 10^{-\frac{3}{20}} w(0) \triangleq w'_k$, then $\tau'_k = Q/(2f_k)$. This definition allows the comparison of Q for methods with different window shapes. Note however, that a similar measure of Q can be formulated in the frequency domain.

In the proposed approach the first step is to design an IIR filterbank that accomplishes the constant Q behavior. Then, a TV IIR filter is devised based on the poles of the filterbank. Finally a fine tuning is performed to improve the steadiness of the Q value for the TV IIR filter. In the following section, this procedure is described in detail.

3.3.1 Proposed design

Following the definition of Q in time, the poles of the IIR filterbank can be calculated from equation 7 as the solution of a second order polynomial: $(2w'_k - \cos(\tau'_k) - 1)p_k^2 + (2 + 2\cos(\tau'_k) - 4w'_k \cos(\tau'_k))p_k + 2w'_k - \cos(\tau'_k) - 1 = 0$.

Then, a simple and effective design of the TV IIR filter consists in choosing for each frequency bin the corresponding pole of the IIR filterbank, that is $p[n] = p_k$, with $k = n$. The Q factors obtained with this approach are close to the desired constant value but with a slight linear drift. This result shows that the slow variation of the LTV system allows an approximation by a single TV IIR with a little deviation that can be easily compensated by adding the same slope to the desired Q value at each bin. Figure 3 shows the Q curve for the original and compensated designs.

Another design consideration is that for low frequencies a constant Q would imply a longer window support than the frame time. It becomes necessary to limit the time τ'_k to a maximum time τ_{max}, such that $2\,\tau_{max}$ is smaller than the frame time. This limitation of τ'_k to a maximum value must be done in a smooth way. Let $\bar{\tau}'_k$ be a new variable that represents the result of saturating τ'_k. The transition can be implemented with a hyperbola whose asymptotes are $\bar{\tau}'_k = \tau'_k$ and $\bar{\tau}'_k = \tau_{max}$, so that $(\bar{\tau}'_k - \tau_{max})(\bar{\tau}'_k - \tau'_k) = \delta$,

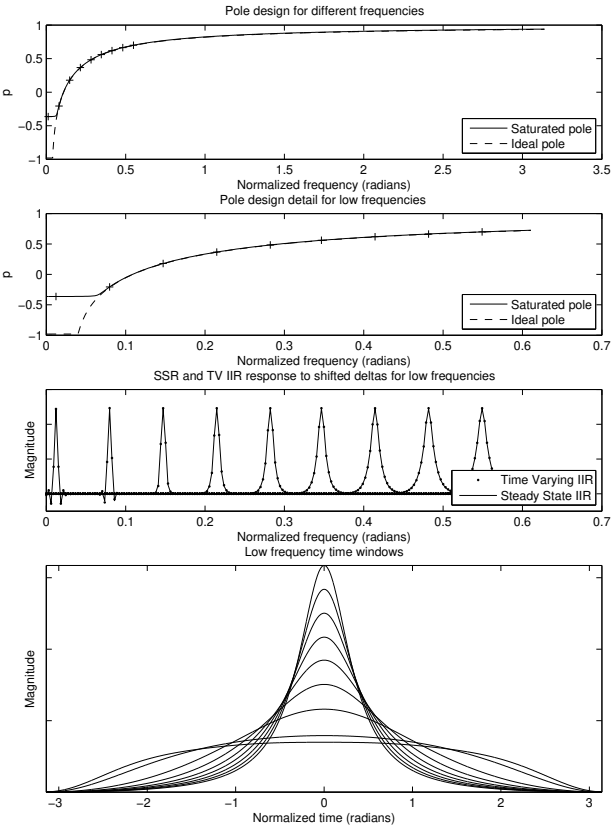

Figure 2. Detail of poles design. Pole locations for the ideal and saturated design. Impulse responses at low frequencies for the TV IIR and the Steady State, along with corresponding TV IIR time windows.

where δ is a constant that determines the smoothness of the transition.

The selection of τ_{max}, affects the behavior of the transform in low frequencies. Choosing a small τ_{max} compared to the frame time gives poor frequency resolution. On the contrary, if τ_{max} is set to a value close to the frame time, a better resolution is expected, but some distortion appears. This is because the time windows get close to a rectangular window for low frequencies. The spectrum of these windows has big side lobes, introducing Gibbs oscillations in the representation. Additionally, as a time window for low frequency approaches to a rectangular shape, its response to an impulse vanishes more slowly, so it becomes necessary to calculate the response for some negative frequency bins, adding extra complexity. In practice it is reasonable to choose an intermediate value of τ_{max}, e.g. $\tau_{max} \approx 0.7\pi$, such that only for very low frequencies the transform exhibits non constant Q. Figure 2 shows details of the described poles design.

3.3.2 TV IIR filtering and zero-padding in time

It is common practice to work with a higher sampling frequency of the spectrum, typically obtained by zero-padding in time. In this case the TV IIR filter design changes, as the signal support becomes $(-\tau_1, \tau_1]$ with $0 < \tau_1 < \pi$. Then, the discontinuity to be avoided at the ends of the frame ap-

```
p = design_poles(NFFT,Q);
X = fft(fftshift(s));
Y'(1) = X(1);
for i = 2:NFFT/2
    Y'(n) = X(n-1) + X(n) + p(n)Y'(n-1);
end
Y(n) = Y'(NFFT/2);
for i = NFFT/2-1:-1:1
    Y(n) = Y'(n+1) + Y'(n) + p(n)Y(n+1);
end
```

Table 1. Pseudocode of the TV IIR filter. First, the poles and normalization factor are designed given the number of bins (`NFFT`) and the Q value. Then the FFT of the signal frame `s` is computed after centering the signal at time 0. Finally the forward-backward TV IIR filtering is performed for that frame.

pears at $\pm\tau_1$, so a couple of zeros at $\pm\tau_1$ have to be placed instead of the zero at π. Window properties outside this support are irrelevant, as windowed data values are zero. The design of poles has to take into account the new zeroes and the time re-scaling, but windows with similar properties are obtained.

3.3.3 Implementation

The method implementation [4] is rather simple, as can be seen in the pseudocode of Table 1. A function to design the poles is called only once and then the forward-backward TV IIR filtering is applied to the DFT of each signal frame. The proposed IIR filtering applies a window centered at time 0, so the signal frame has to be centered before the transform. To avoid transients at the ends, the filtering should be done circularly using a few extra values of the spectrum as prefix and postfix. Their lengths can be chosen so as truncation error lies below a certain threshold, for instance 60 dB.

4. METHODS COMPARISON

4.1 Frequency scale

Depending on the context of the music analysis application different frequency grids may be preferred. To this respect, the efficient CQT method can be designed for any arbitrary frequency spacing. On the contrary, the MR FFT and the IIR CQT are constrained to a linear frequency scale because they rely on the DFT. This spacing typically implies an oversampling at high frequencies to conform with the minimum spacing at low frequencies.

4.2 Effective quality factor

The analyzed methods have different flexibility to define an arbitrary Q at each frequency. The efficient CQT offers the freedom to set any possible Q for every bin. The MR FFT allows choosing the resolution for every bin from a reduced set not enabling an arbitrary Q. On the other hand,

[4] The complete code is available at `http://iie.fing.edu.uy/~pcancela/iir-cqt`.

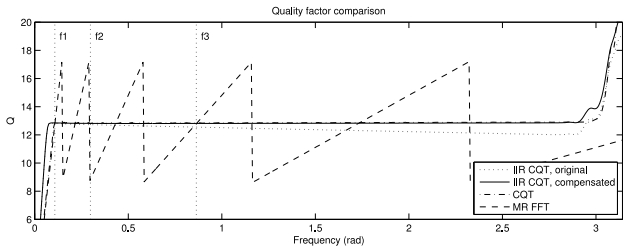

Figure 3. Comparison of the effective Q for a target value of 12.9 given the definition of 3.3. This value gives 34 cycles within the window, as commonly used in the CQT.

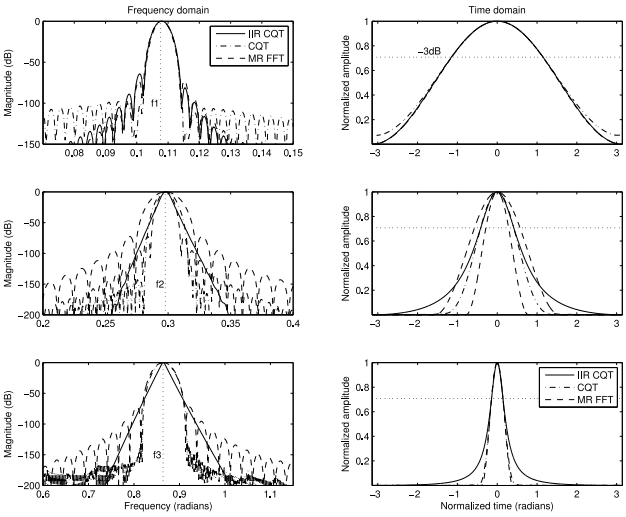

Figure 4. Windows comparison at frequencies f_1, f_2 and f_3 for the different methods. At f_1 and f_3 the three methods have the same Q, while at f_2 the MR FFT can not achieve the desired Q. For this reason, the two nearest MR FFT windows are considered at f_2. CQT and MR FFT are computed using a Hamming and Hann windows respectively.

the TV IIR filter allows any Q value for any frequency but with the constraint that it evolves slowly with frequency. This holds particularly well in the case of a constant Q transform, so the IIR CQT can give any constant Q with a fairly simple design. Figure 3 shows the obtained Q with the different methods. It can be observed that the MR FFT has a bounded Q due to the resolution quantization.

4.3 Windows properties

The spectral and temporal characteristics of windows at three different frequencies are shown in Figure 4 for each method. At frequency f_1, IIR CQT time window behaves like a Hann window. For lower frequencies it exhibits a flatter shape to extend the range of constant Q (see Figure 2). For higher frequencies, the main lobe of the obtained windows has a steeper drop up to -50 dB compared to a conventional Hann or Hamming window. As a counterpart, time resolution is slightly diminished. Note that the selected drop value in the definition of Q sets the location in this compromise.

4.4 Computational complexity

The three algorithms are compared based on the number of real floating point operations performed in mean for each frequency bin. All of them compute the DFT of a non windowed frame, so these operations are not considered.

The number of operations in the efficient CQT depends on the length of the frequency kernels. This length varies with Q and is different for different frequency bins. For the Q and threshold values used in Figures 3 and 4 (Q_{CQT} = 34, Q = 12.9, th = 0.0054), `NFFT` = 2048 and $f_s = 44100$ Hz, the frequency kernel length varies from 1 to 57 coefficients, which implies a mean number of 27 real multiplications and 27 real additions. This result depends on the threshold and inversely on Q. The MR FFT takes advantage of the hierarchical implementation of the FFT to compute the transform, so the windowing in the frequency domain needs only 3 complex sums and 2 multiplications for each bin. The total number of real floating point operations is then, 4 multiplications and 6 additions. The IIR CQT involves a forward and backward IIR filtering with a variable real pole and a zero, followed by a real normalization (see Table 1 for a pseudocode). As the frequency components are complex values, the necessary number of real operations to compute each bin is 6 multiplications and 8 additions (plus a negligible number of extra operations due to the circularly filtering approximation).

5. APPLICATIONS AND RESULTS

Finally, two different examples of the spectral analysis of polyphonic music using the proposed IIR CQT method are shown in Figure 5 together with conventional spectrograms. As it is expected in a constant Q transform, it can be noticed that singing voice partials with high frequency slope tend to blur in the spectrogram but are sharper in the IIR CQT. This improved time resolution in high frequencies also contributes to define more precisely the note onsets, as can be seen in the second example (e.g. the bass note at the beginning). Moreover, in the low frequency band, where there is a higher density of components, the IIR CQT achieves a better discrimination, due to the fact that its time windows are flatter than typically used windows. At the same time, frequency resolution for the higher partials of notes with a steady pitch is deteriorated.

The proposed IIR CQT method was used as part of the spectral analysis front-end of a melody extraction algorithm submitted to the MIREX Audio Melody Extraction Contest 2008, performing best on Overall Accuracy [5]. Although the constant Q behavior of the spectral representation is just a small component of the algorithm, the results may indicate that the usage of the IIR CQT is appropriate.

6. CONCLUSIONS

In this work a novel method for computing a constant Q spectral transform is proposed and compared with two existing techniques. It shows to be a good compromise between the flexibility of the efficient CQT and the low computational cost of the MR FFT. Taking into account that it was used in the spectral analysis of music with encouraging results and that its implementation is rather simple, it seems to be a good spectral representation tool for audio signal analysis algorithms.

[5] The MIREX 2008 evaluation procedure and results are available at `http://www.music-ir.org/mirex/2008/index.php/Audio_Melody_Extraction`.

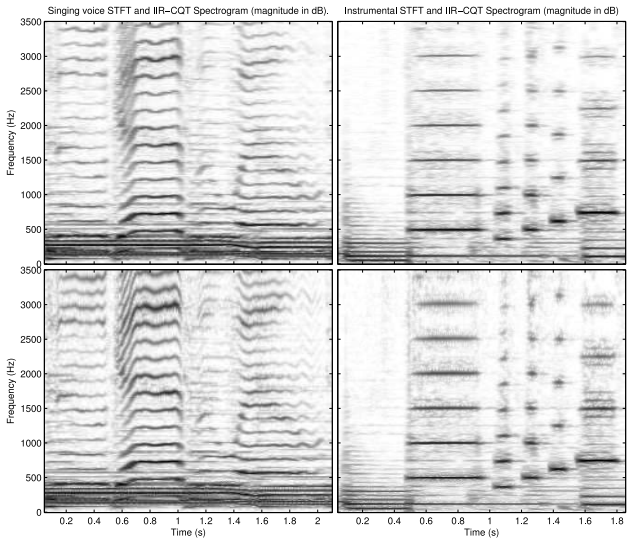

Figure 5. STFT and IIR CQT for two audio excerpts, one with a leading singing voice and the other, instrumental music.

7. REFERENCES

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

[2] K. Dressler, "Sinusoidal Extraction Using and Efficient Implementation of a Multi-Resolution FFT," in *Proceedings of the DAFx-06*, (Montreal, Canada), 2006.

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

[4] K. L. Kashima and B. Mont-Reynaud, "The bounded-Q approach to time-varying spectral analysis," Tech. Rep. STAN-M-28, Stanford University, 1985.

[5] F. C. C. B. Diniz, I. Kothe, S. L. Netto, and L. W. P. Biscainho, "High-Selectivity Filter Banks for Spectral Analysis of Music Signals," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, 2007.

[6] M. Goto, "A Real-time Music Scene Description System: Predominant-F0 Estimation for Detecting Melody and Bass Lines in Real-world Audio Signals," *Speech Communication (ISCA Journal)*, vol. 43, no. 4, pp. 311–329, 2004.

[7] J. S. Prater and C. M. Loeffler, "Analysis and design of periodically time-varying IIR filters, with applications to transmultiplexing," *IEEE Transactions on Signal Processing*, vol. 40, no. 11, pp. 2715–2725, 1992.