A Matlab Toolbox for Efficient Perfect Reconstruction Time-Frequency Transforms with Log-Frequency Resolution

Christian Schörkhuber[1,2], Anssi Klapuri[1,3], Nicki Holighaus[4], Monika Dörfler[5]

[1]*Tampere University of Technology, Tampere, Finland*

[2]*Institute of Electronic Music and Acoustics, University of Music and Performing Arts Graz, Graz, Austria*

[3]*Ovelin Ltd, Helsinki, Finland*

[4]*Acoustics Research Institute, Austrian Academy of Sciences, Vienna, Austria*

[5]*Numerical Harmonic Analysis Group, Faculty of Mathematics, University of Vienna, Vienna, Austria*

Correspondence should be addressed to Christian Schörkhuber (`schoerkhuber@iem.at`)

ABSTRACT

In this paper, we propose a time-frequency representation where the frequency bins are distributed uniformly in log-frequency and their Q-factors obey a linear function of the bin center frequencies. The latter allows for time-frequency representations where the bandwidths can be e.g. constant on the log-frequency scale (constant Q) or constant on the auditory critical-band scale (smoothly varying Q). The proposed techniques are published as a Matlab toolbox that extends [3]. Besides the features that stem from [3] – perfect reconstruction and computational efficiency – we propose here a technique for computing coefficient phases in a way that makes their interpretation more natural. Other extensions include flexible control of the Q-values and more regular sampling of the time-frequency plane in order to simplify signal processing in the transform domain.

1. INTRODUCTION

Time-frequency representations of discrete time domain signals play an important role in audio signal processing and analysis. The short time Fourier transform (STFT) is here the most widely-used tool, although it is generally acknowledged that the linear frequency bin spacing of the discrete Fourier transform (DFT) is not in agreement with the frequency resolution of the human auditory system and the geometric distribution of fundamental frequencies in music.

In [1] a constant-Q transform (CQT) was proposed, where the frequency bins are geometrically spaced and have equal Q-factors, that is, analysis window sizes increase towards lower frequencies. However, the lack of efficient algorithms for computing the CQT as well as the absence of an inverse transform hindered the widespread use of this transform in music and speech signal processing for almost two decades.[1] Addressing these shortcomings, in [2] a CQT toolbox was proposed allowing for efficient computation of CQT coefficients as well as reasonable quality reconstruction (around 55 dB SNR) of the time domain signal from its transform coefficients.

In [3] a constant-Q transform toolbox has been proposed that further increases the efficiency of computing the transform and allows for perfect reconstruction of the time domain signal. The proposed transform in [3] is a special case of the non-stationary Gabor transform (NSGT) [4, 5, 6], namely the constant-Q NSGT (CQ-NSGT) (see Sec. 3).

While exhibiting the aforementioned advantages compared to the method proposed in [2], the implementation of the CQ-NSGT in [3] involves some drawbacks from the viewpoint of practical applications, such as a dispersive time sampling grid and non-intuitive interpretation of the obtained phase values. Furthermore, a gen-

[1] CQT is essentially a wavelet transform with high frequency resolution (typically 10–100 bins per octave). This renders classical wavelet transform techniques inadequate for computing the CQT.

eral problem of constant-Q transforms is that the time-domain windows get unreasonably long at very low frequencies. Some of these issues have recently been addressed in [7, 8, 9]. However, to the best of our knowledge there is currently no implementation available that solves all the above problems.

In this paper we present a Matlab toolbox for perfect-reconstruction, variable-Q transforms with geometrically spaced frequency bins and smoothly varying Q-factors. This toolbox is a variation of the toolbox presented in [3], allowing for more intuitive interpretation of the transform coefficient phase values and time-aligned sampling of the time-frequency plane.

The paper is organized as follows. In Sec. 2, we define the basic constant-Q transform. In Sec. 3, we describe the computation of CQT using the algorithm proposed in [3]. In Sec. 4, we describe how the transform coefficient phases can be computed. In Sec. 6, we generalize the transform to allow the Q-factors of the frequency bins to be a linear function of the center frequencies, instead of being directly proportional to the center frequencies. In Sec.7, we describe a technique for aligning the transform coefficients in time.

2. CONSTANT-Q TRANSFORM

The CQT representation $X(k,n)$ of a discrete time domain signal $x(n)$ is defined as

$$X(k,n) = \sum_{m=0}^{N} x(m) a_k^*(m-n), \tag{1}$$

where k and n denote frequency and time indices, respectively, N is the length of the input signal $x(n)$ and the atoms $a_k^*(t)$ are the complex conjugated modulated localization functions (window functions) defined by

$$a_k(m) = g_k(m) e^{i2\pi m f_k/f_s}, \quad m \in \mathbb{Z}, \tag{2}$$

with the zero-centered window function $g_k(m)$ and the bin center frequency f_k, the sampling rate f_s and $i = \sqrt{-1}$. The center frequencies are geometrically spaced such that

$$f_k = f_0 2^{\frac{k}{b}}, \quad k = 0, 1, \ldots, K-1 \tag{3}$$

where b determines the number of frequency bins per octave, f_0 is the lowest frequency analysed and K is the overall number of frequency bins. In CQT, the Q-factor (ratio of center frequencies to bandwidths) is defined to be constant, hence the support of the window $g_k(m)$ (time range where it has significant non-zero values) is inversely proportional to f_k.

3. CQT BY SIMULATING A FILTERBANK IN FREQUENCY DOMAIN

In this section, we outline the algorithm proposed in [3] for computing the CQT and its inverse transform from a filterbank point of view.

Assuming $g_k(m)$ to be a symmetric function about $m = 0$ we note that $a_k^*(m) = a_k(-m)$ an rewrite (1) to

$$X(k,n) = \sum_{m=0}^{N} x(m) a_k(n-m) \tag{4}$$

$$= [x * a_k](n) \tag{5}$$

$$= {F_N}^{-1} \left[(F_N x)(F_N a_k) \right](n), \tag{6}$$

where $*$ denotes the convolution and F_N is the N-point discrete Fourier transform (DFT) operator. Denoting by $\hat{x} = F_N x$ the N-point DFT sequence of $x(n)$ and with $\hat{a_k} = F_N a_k$ the N-point DFT sequence of $a_k(n)$ we write

$$X(k,n) = c_k(n) = [{F_N}^{-1}(\hat{x}\hat{a_k})](n) \tag{7}$$

$$= [{F_N}^{-1}\hat{c_k}](n) \tag{8}$$

Hence, the CQT coefficients as defined in (1) can be equivalently computed by means of a fast convolution (multiplication in the DFT domain). However, this would still require K N-point IDFT operations, which is not efficient computationally.

The natural approach to reducing the complexity of the transform is to evaluate $X(k,n)$ not for every $n \in 0, 1, ..., N-1$ but only for every $n \in 0, H_k, 2H_k, ..., \frac{N-1}{H_k}$ where H_k is referred to as the analysis hop size for each frequency bin k. This can also be understood as subsampling each output $c_k(n)$ of the K-channel filterbank with a sampling rate $f_s^k = f_s/H_k$, where f_s is the original sampling rate of the input signal. From bandpass sampling theory we know that a bandpassed analytical signal can be perfectly reconstructed from its subsampled version if the frequency responses $\hat{a_k}$ have compact support with upper and lower bounds f_k^u and f_k^l, respectively, and $f_s^k \geq B_k$ with $B_k = f_k^u - f_k^l$ (in the context of frame theory this is referred to as the *painless* case [10]). That is, a first step to obtain perfect reconstruction of the transform

in [3] (see Sec. 5) is to choose the window functions g_k, such that they have compact support in the *frequency domain* as opposed to the more traditional approach of g_k having compact support in time domain.

To reduce the computational complexity of evaluating (6) followed by subsampling in time domain, subsampling of c_k can be realized by periodization of $\hat{c}_k$ as follows:

To mimic time-domain subsampling in the frequency domain, one has to map the entire spectrum ranging from $-\frac{f_s}{2}$ to $+\frac{f_s}{2}$ into the frequency interval $]-f_s^k/2, f_s^k/2]$. In the toolbox presented in [3] this step is performed by shifting down each $\hat{c}_k$ by the frequency f_k such that all $\hat{c}_k$ are centered around zero frequency. Subsequently all DFT bins outside the range $]-f_s^k/2, f_s^k/2]$ are discarded and the transform coefficients are obtained by applying an IDFT operation to the remaining DFT coefficients.[2]

4. OBTAINING CQT PHASES

The algorithm described in Sec. 3 leads to valid transform coefficients, but the employed subsampling procedure is not equivalent to time-domain subsampling and therefore the obtained transform coefficients are not the same as those obtained by evaluating (1). More specifically, the absolute values of the coefficients are the same, but their phase values differ from those obtained using (1).

In this section we will describe how the implementation of [3] can be modified to exactly reproduce the transform coefficients obtained from evaluating (1).

Subsampling in the frequency domain as outlined in the previous section is performed by reducing the number of DFT bins. As long as $f_s^k \geq B_k$ no *harmful* aliasing[3] is introduced and the original signal can be easily reconstructed (see section 5). However, the transform coefficients thus obtained do not necessarily correspond to the coefficients obtained when subsampling is performed in the time domain. To exactly mimic time-domain subsampling in the frequency domain, all non-zero spectral components in the range between $-f_s/2$ and $f_s/2$ have to be mapped to the frequency range $]-f_s^k/2, f_s^k/2]$ with the mapping function

$$\mathrm{M}(f, f_s^k) = f - \left\lfloor \frac{f}{f_s^k} \right\rfloor f_s^k, \tag{9}$$

where f is the original frequency, $\mathrm{M}(f, f_s^k)$ is the image frequency after subsampling and $\lfloor \cdot \rfloor$ denotes rounding towards negative infinity. The mapping function (9) can be easily verified by envisioning that the main effect of sampling a continuous time domain signal is that the entire spectrum gets replicated at each integer multiple of the sampling rate.

In [3], on the other hand, each $\hat{c}_k$ is shifted to the baseband, i.e. a different mapping function is used. This leads to valid transform coefficients, however, the interpretation of their phase values is somewhat less intuitive. In Figure 1 the mapping process is sketched for one frequency channel k. The transform coefficients $\hat{c}_k = \hat{x}\hat{a}$ sampled at the rate f_s in the upper panel are mapped to the range $]-f_s^k/2, f_s^k/2]$ in the lower panel by applying zero-centered mapping and mapping with $\mathrm{M}(f, f_s^k)$, respectively. It can be observed, that the mapping function $\mathrm{M}(f, f_s^k)$ generates a circularly shifted spectrum where the shift is given by $\mathrm{M}(f_k, f_s^k)$. If the center frequencies f_k are rounded to coincide with a DFT bin (which is a negligible constraint if sufficiently long input signals are considered), then the desired shift in DFT bins $s = \mathrm{M}(f_k, f_s^k)\frac{N}{f_s}$ is integer valued.

In Figure 2 exemplary transform coefficients of one frequency channel k are depicted for three different implementations: $c_k(n)$ are the transform coefficients at the original rate (without subsampling), $c_k^2(\tilde{n})$ and $c_k^1(\tilde{n})$ are the subsampled coefficients with and without applying a circular shift of s DFT bins in the frequency domain, respectively. It can be observed that after the circular shift has been applied in the frequency domain, the transform coefficients exactly subsample the original output signal of filter channel k whereas otherwise only the coefficients absolute values coincide.

Hence, with the above modification the CQ-NSGT [3] yields transform coefficients that are identical to those obtained from brute-force evaluation of (1) or from previous CQT implementations [1, 2].[4]

[2] Note that this is only true for the painless case. Generally speaking, in [3, 7] all bins from f_k^l to f_k^u are periodized with respect to f_s^k which also allows for a non-painless setup (where $f_s^k < B_k$). However, here we only consider the painless case.

[3] Aliasing is introduced as soon as $f_s^k < f_k^u$, however for one-sided bandpass signals (analytic signals) *harmful* aliasing is not introduced as long as $f_s^k \geq B_k$, i.e spectral components do not overlap due to aliasing.

[4] Provided that the window functions $g_k(n)$ and the hop sizes H_k are the same of course. Note that the hop sizes in the method used here are not necessary integer multiples of the sampling interval.

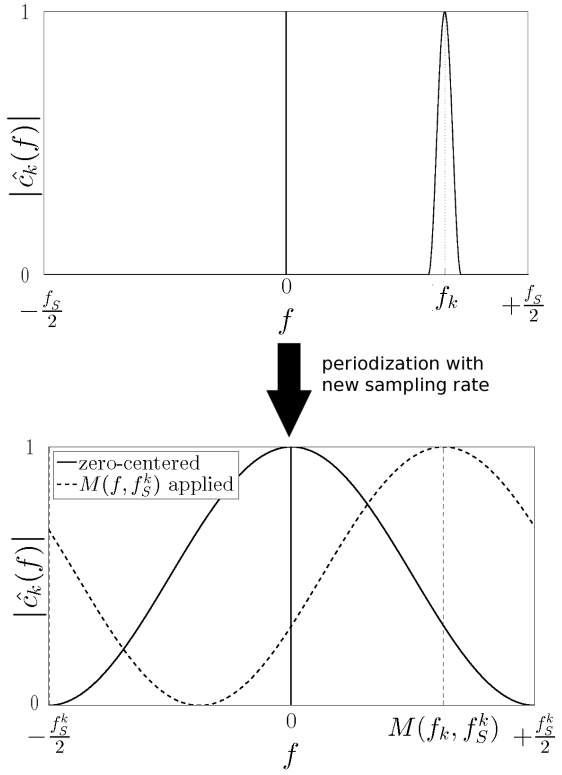
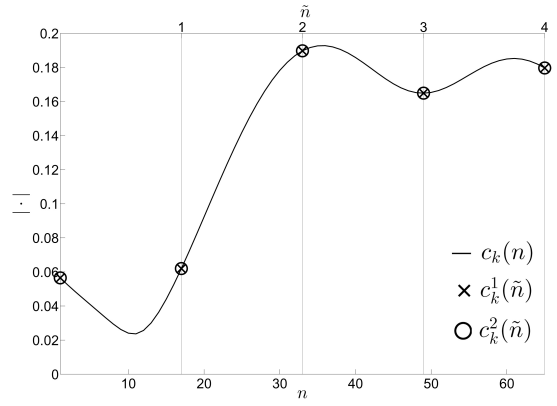
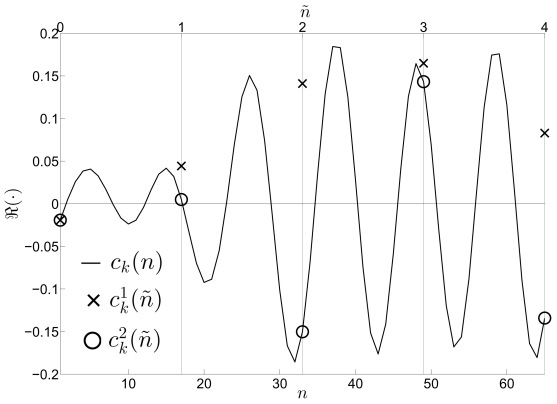


Fig. 1: Exemplary mapping process for one filter channel k ($f_s^k = B_k$). Upper panel: transform coefficients $\hat{c}_k$ sampled at the original sampling rate f_s. Lower panel: mapped transform coefficients according to the new sampling rate f_s^k using zero-centered mapping (solid line) and mapping with $\mathrm{M}(f, f_s^k)$ (dotted line), respectively.



(a) Absolute values of transform coefficients



(b) Real parts of transform coefficients

Fig. 2: Exemplary transform coefficients of one particular frequency channel k. $c_k(n)$ is the channel output without subsampling, $c_k^1(\tilde{n})$ is the subsampled channel output as implemented in [3] and $c_k^2(\tilde{n})$ is the subsampled channel output after applying the mapping function $M(f, f_s^k)$.

5. PERFECT RECONSTRUCTION

In [3] it has been shown that the transform yields perfect reconstruction by using *dual frames* $\tilde{a}_k$ for synthesis. If the analysis atoms a_k have compact support in the frequency domain (*painless case*) and their joint support covers the entire frequency plane (necessary criterion for the set of analysis atoms being a *frame* [11]), computation of synthesis atoms $\tilde{a}_k$ is straightforward (see [3, 6] for theoretical background and proof). To meet the latter condition, two additional filters are introduced to cover the frequency ranges from zero to f_0 and from f_{K-1} to the Nyquist frequency. In [9] it has been shown that in some cases perfect reconstruction of the NSGT can be achieved even when $f_S^k < B_k$ (*non-painless case*) using iterative algorithms (adapted *conjugate gradients* algo-

rithm [12, 13]) to find a dual frame that accounts for the introduced harmful aliasing. The implementation of these concepts is beyond the scope of this contribution.

6. VARIABLE-Q

As discussed in Introduction, CQT has several advantages over STFT when analysing music signals. However, one considerable practical drawback is the fact that the analysis/synthesis atoms get very long towards lower frequencies. This is unreasonable both from a perceptual viewpoint and from a musical viewpoint. Auditory filters in the human auditory system are approximately constant-Q only for frequencies above 500 Hz and smoothly approach a constant bandwidth towards lower frequencies. Accordingly, music signals generally do not contain closely spaced pitches at low frequencies, thus the Q-factors (relative frequency resolution) can safely be reduced towards lower frequencies, which in turn improves the time resolution. This has been addressed e.g. in [9], where the authors have proposed a so-called *ERBlet transform*. In the ERBlet transform, the bin bandwidths and center frequencies correspond to the equivalent rectangular bandwidths (ERB) [14] and their corresponding frequency distribution, respectively. In [9], also a reference implementation is provided. Similarly, in [8] the use of a filter channel distribution according to the Bark scale is outlined.

In the toolbox proposed here, we maintain the geometrical bin spacing from the constant-Q approach with the parameter b defining the number of frequency bins per octave. That is done for the sake of clarity and convenience when processing music signals. However, we introduce an additional parameter γ that allows for smoothly decreasing the Q-factors of the bins towards low frequencies. We define the bandwidth[5] B_k of filter channel k as

$$B_k = \alpha f_k + \gamma, \tag{10}$$

where

$$\alpha = 2^{\frac{1}{b}} - 2^{-\frac{1}{b}} \tag{11}$$

is determined by the number of bins per octave, b. In Figure 3 the bandwidth B_k is plotted over center frequencies f_k and different values for γ. Special cases are $\gamma = 0$ (constant-Q) and $\gamma = \Gamma$ where the bandwidths equal

[5]For the sake of readability, here we use the term bandwidth to denote the overall support of the analysis atoms in frequency domain. This is in contrast to the commonly used -3 dB bandwidth.

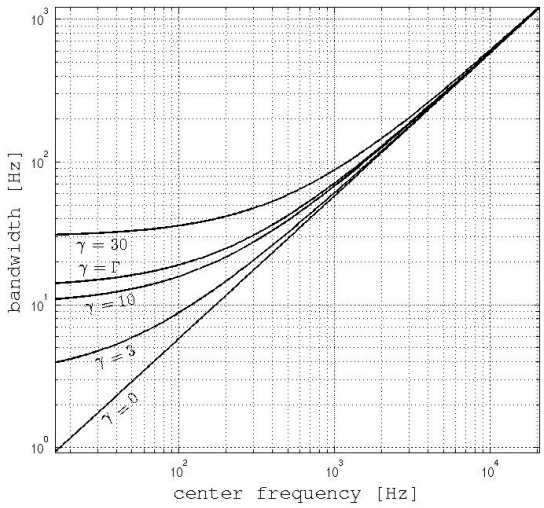


Fig. 3: Filter bandwidth over center frequencies and different values for γ (resolution $b = 24$).

a constant fraction of the ERB critical bandwidth [14]. Here

$$\Gamma = \frac{24.7}{0.108}\alpha \tag{12}$$

such that

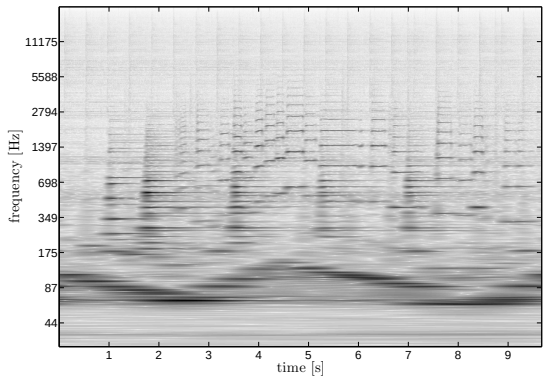
$$B_k = \frac{\alpha}{0.108}\mathrm{ERB}. \tag{13}$$

In Figure 4 the time-frequency representations of a music signal are depicted for different values of γ. It can be observed that larger values of γ increase the time resolution at lower frequencies.

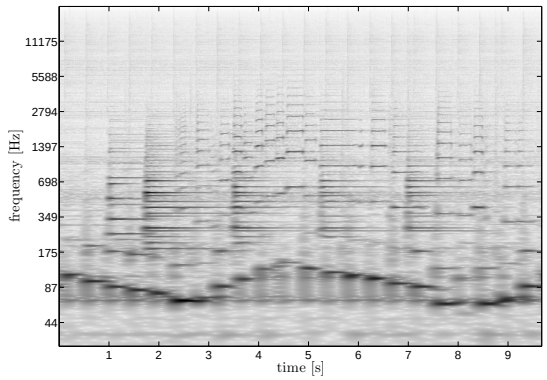
7. TIME-ALIGNED COEFFICIENTS

The lowest redundancy of the CQ-NSGT is obtained when all filter channel outputs (coefficients corresponding to a certain CQT bin) are critically sampled. This implies that hop sizes between two sampling points along time are distinct for each frequency channel, thus the transform coefficients cannot be presented as a matrix. However, it is often desirable that the sampling points are aligned in time (e.g. in order to perform frequency translation [15] or to apply algorithms that rely on a time-frequency matrix such as non-negative matrix factorization [16]).

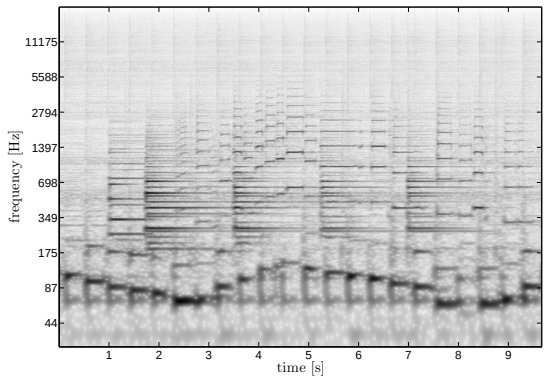
Temporal alignment can be easily achieved by applying a common subsampling factor for all frequency bins



(a) $\gamma = 0$ (constant-Q case)



(b) $\gamma = \Gamma = 6.6$ (constant fraction of ERB)



(c) $\gamma = 20$

Fig. 4: Proposed time-frequency representation of a music excerpt containing upright bass, drums, piano and trumpet using $b = 48$ and different values for γ.

$k \in 1, ..., K$. That is, only the highest frequency channel is critically sampled and all other channels are subsampled with the same rate (we refer to this as *full rasterization*). Obviously this considerably increases the redundancy of the representation, especially when analysing the signal up to very high frequencies (which for some applications might not be necessary). To address this issue, in the proposed toolbox *piecewise rasterization* of the representation is provided as an option. Here the hop sizes for all frequency channels are rounded down to power-of-two multiples of the the highest frequency bin hop size H_K. That is,

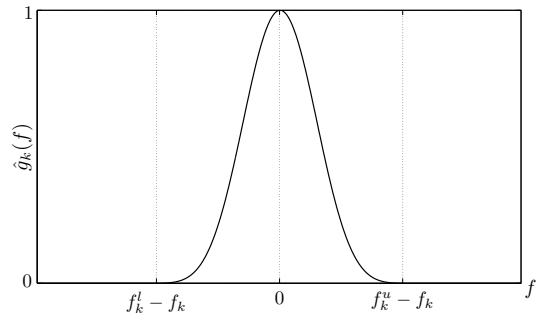
$$H_k = H_K^c \cdot 2^p \leq H_k^c \tag{14}$$

where H_k^c denotes the hop size for frequency channel k in the critically sampled case and $p \in \mathbb{N}_0$ is chosen such that H_k is close to H_k^c.[6]

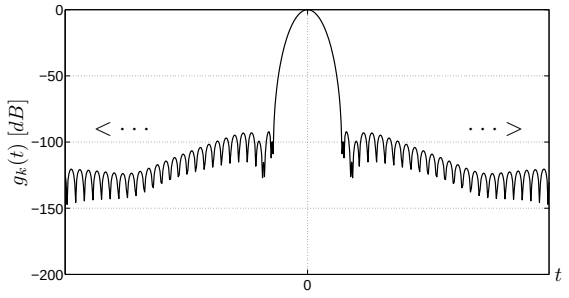
8. CHOICE OF THE WINDOW FUNCTION

In the proposed toolbox we use time-frequency atoms with compact support in the frequency domain and maximum subsampling factors (minimum sampling rates) such that no harmful aliasing occurs, i.e. bandpassed spectral components do not overlap after subsampling. In the context of frame theory this is referred to as the painless case where perfect reconstruction using dual frames for synthesis is straightforward. However, compact support in frequency domain implies infinite filter impulse responses (IIR filter) given by the inverse Fourier transforms of the atoms $\hat{a}_k$. Two drawbacks arise from this implementation. Firstly, an impulse in time domain will exhibit sidelobes along time in the time-frequency representation as opposed to the familiar spectral sidelobes of sinusoidal components. Secondly, the IIR property of the transform excludes a realtime implementation. To illustrate this, an exemplary window function g_k where $\hat{g}_k$ is a Blackman-Harris window is depicted in Figure 5. By choosing a proper window function for filter construction in the frequency domain (e.g. Blackman-Harris window), however, temporal sidelobes do not pose a serious problem in practical applications. Furthermore, as we do not aim for a realtime implementation in this contribution we can savely use zero-phase IIR filters. However, a bounded-delay constant-Q implementation has been proposed in [7] (dubbed *sliCQT*) where the signal is processed in overlapping time slices. The effects of temporal aliasing in this implementation

[6] For the constant-Q case ($\gamma = 0$) this leads to an octavewise rasterization, similar to that in [2].



(a) Window function in the frequency domain (compact support).



(b) Window function in the time domain (infinite support).

Fig. 5: Exemplary window function g_k in time and frequency domains.

are mitigated by hard-limiting the window sizes towards lower frequencies and zero-padding.

9. CONCLUSION

In this paper a Matlab toolbox for perfect reconstruction time-frequency transforms with logarithmic frequency bin spacing and smoothly varying Q-factors has been proposed. The toolbox is a variation of the toolbox proposed in [3] providing intuitive interpretation of the transform coefficients phase values, time-aligned sampling of the time-frequency plane and frequency bin bandwidths which are defined by a linear function allowing e.g. constant-Q or fraction-of-critical-band bandwidths. The toolbox can be downloaded from *www.cs.tut.fi/sgn/arg/CQT/*.

10. REFERENCES

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

[2] C. Schörkhuber and A. Klapuri, "Constant-Q transform toolbox for music processing," in *Proc. Sound and Music Computing Conference (SMC)*, 2010.

[3] G.A. Velasco, N. Holighaus, M. Dörfler, and T. Grill, "Constructing an invertible constant-Q transform with nonstationary gabor frames," in *Proc. Digital Audio Effects (DAFx-11)*, 2011.

[4] F. Jaillet, *Représentation et traitement temps-fréquence des signaux audionumériques pour des applications de design sonore*, Ph.D. thesis, Université de la Méditerranée - Aix-Marseille, 2005.

[5] F. Jaillet, P. Balazs, M. Dörfler, et al., "Nonstationary gabor frames," in *SAMPTA'09, International Conference on Sampling Theory and Applications*, 2009.

[6] P. Balazs, M. Dörfler, F. Jaillet, N. Holighaus, and G. Velasco, "Theory, implementation and applications of nonstationary gabor frames," *Journal of Computational and Applied Mathematics*, pp. 236:1481–1496, 2011.

[7] N. Holighaus, M. Dorfler, G.A. Velasco, and T. Grill, "A framework for invertible, real-time constant-Q transforms," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 4, pp. 775–785, 2013.

[8] Gianpaolo Evangelista, Monika Dörfler, and Ewa Matusiak, "Arbitrary phase vocoders by means of warping," *Musica/Tecnologia*, vol. 7, pp. 91–118, 2013.

[9] T. Necciari, P. Balazs, N. Holighaus, and P. Sondergaard, "The erblet transform: An auditory-based time-frequency representation with perfect reconstruction," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013.

[10] I. Daubechies, A. Grossmann, and Y. Meyer, "Painless nonorthogonal expansions," *Journal of Mathematical Physics*, vol. 27, pp. 1271, 1986.

[11] O. Christensen, *An introduction to frames and Riesz bases*, Birkhauser, 2003.

[12] K. Grochenig, "Acceleration of the frame algorithm," *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3331–3340, 1993.

[13] L.N. Trefethen and D. Bau III, *Numerical linear algebra*, Number 50. Siam, 1997.

[14] B.R. Glasberg, "Derivation of anditory filter shapes from notched-noise data," *Hearing Research*, pp. 103–138, 1990.

[15] C. Schörkhuber, A. Klapuri, and A. Sontacchi, "Audio pitch shifting using the constant-Q transform," *Journal of the Audio Engineering Society*, vol. 61, no. 7/8, pp. 562–572, 2013.

[16] D Seung and L Lee, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, pp. 556–562, 2001.