# A Framework for Invertible, Real-Time Constant-Q Transforms

Nicki Holighaus, Monika Dörfler, Gino Angelo Velasco, and Thomas Grill

***Abstract*—Audio signal processing frequently requires time-frequency representations and in many applications, a non-linear spacing of frequency bands is preferable. This paper introduces a framework for efficient implementation of invertible signal transforms allowing for non-uniform frequency resolution. Non-uniformity in frequency is realized by applying *nonstationary Gabor frames* with adaptivity in the frequency domain. The realization of a perfectly invertible *constant-Q* transform is described in detail. To achieve real-time processing, independent of signal length, slice-wise processing of the full input signal is proposed and referred to as *sliCQ transform*. By applying frame theory and FFT-based processing, the presented approach overcomes computational inefficiency and lack of invertibility of classical constant-Q transform implementations. Numerical simulations evaluate the efficiency of the proposed algorithm and the method's applicability is illustrated by experiments on real-life audio signals.**

***Index Terms*—Audio signals, constant-Q, Gabor frames, real-time, time-frequency dictionary.**

## I. Introduction

ANALYSIS, synthesis and processing of sound is commonly based on the representation of audio signals by means of time-frequency dictionaries. The short-time Fourier transform (STFT), also referred to as *Gabor transform*, is a widely used tool due to its straightforward interpretation and FFT-based implementation, which ensure efficiency and invertibility [7], [15]. STFT features a uniform time and frequency resolution and a linear spacing of the time frequency bins.

In contrast, the constant-Q transform (CQT), originally introduced in [22] and in music processing by J. Brown [2], provides a frequency resolution that depends on geometrically spaced center frequencies of the analysis windows. In particular, the Q-factor, i.e. the ratio of center frequency to bandwidth of each window, is constant over all frequency bins; the constant Q-factor leads to a finer frequency resolution in low frequencies whereas time resolution improves with increasing frequency.

Manuscript received June 08, 2012; revised September 22, 2012 and December 06, 2012; accepted December 06, 2012. Date of publication December 20, 2012; date of current version January 18, 2013. This work was supported by the WWTF project Audio-Miner (MA09-024), the Austrian Science Fund (FWF):[T384-N13], and the EU FET Open grant UNLocX (255931). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Emmanuel Vincent.

N. Holighaus is with the Acoustics Research Institute, Austrian Academy of Sciences, A-1040, Vienna, Austria (e-mail: nicki.holighaus@univie.ac.at).

M. Dörfler is with the Numerical Harmonic Analysis Group, Faculty of Mathematics, University of Vienna, A-1090 Vienna, Austria.

G. A. Velasco is with the Institute of Mathematics, College of Science, University of the Philippines, Diliman, Quezon City 1101, Philippines.

T. Grill is with the Austrian Research Institute for Artificial Intelligence, A-1010 Vienna, Austria.

Digital Object Identifier 10.1109/TASL.2012.2234114

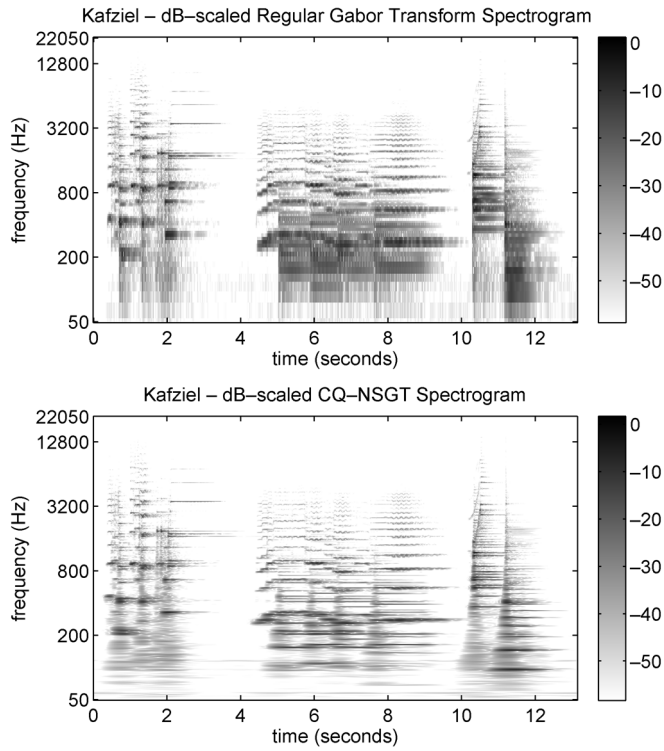



Fig. 1. Time-frequency representations on a logarithmically scaled frequency axis: STFT spectrogram (top) and constant-Q NSGT spectrogram (bottom).

This principle makes the constant-Q transform well-suited for audio data, since it better reflects the resolution of the human auditory system than the linear frequency-spacing provided by the FFT, cf. [20] and references therein. Furthermore, musical characteristics such as overtone structures remain invariant under frequency shifts in a constant-Q transform, which is a natural feature from a perception point of view. In speech and music processing, perception-based considerations are important, which is one of the reasons why CQTs, due to their previously discussed properties, are often desirable in these fields. An example of a CQ-transform, obtained with our algorithm, is shown in Fig. 1.

The principal idea of CQT is reminiscent of wavelet transforms, compare [19]. As opposed to wavelet transforms, the original CQT is not invertible and does not rely on any concept of (orthonormal) bases. On the other hand, the number of bins (frequency channels) per octave is much higher in the CQT than most traditional wavelet techniques would allow for. Partly due to this requirement, the computational efficiency of the original transform as well as its improved versions, cf. [3], may often be insufficient. Moreover, the lack of invertibility of existing

CQTs has become an important issue: for some desired applications, such as extraction and modification, e.g. transposition, of distinct parts of the signal, the unbiased reconstruction from analysis coefficients is crucial. Approximate methods for reconstruction from constant-Q coefficients have been proposed before, in particular for signals which are sparse in the frequency domain [5] and by octave-wise processing in [18].

In the present contribution, we are interested in inversion in the sense of *perfect reconstruction*; to this end, we investigate a new approach to constant-Q signal processing. The presented framework has the following core properties:

1 Relying on concepts from frame theory, [15], we suggest the implementation of a constant-Q transform using the nonstationary Gabor transform (NSGT), which guarantees perfect invertibility. This perfectly invertible constant-Q transform is subsequently called *constant-Q nonstationary Gabor transform* (CQ-NSGT).
2 We introduce a preprocessing step by *slicing* the signal to pieces of (usually uniform) finite length. Together with FFT-based methods, this allows for bounded delay and results in linear processing time. Thus, our algorithm lends itself to real-time processing and the resulting transform is referred to as *sliced constant-Q transform* (sliCQ).

NSGTs, introduced in [1], [11], generalize the classical sampled short-time Fourier transform or Gabor transform [10], [15]. They allow for fast, FFT-based implementation of both analysis and reconstruction under mild conditions on the analysis windows. The CQ-NSGT was first presented in [21]; the frequency-resolution of the proposed CQ-NSGT is indistinguishable from that of the CQT, cf. Fig. 1 for an example.

The main drawback of the CQ-NSGT is the inherent necessity to obtain a Fourier transform of the entire signal prior to actual processing. This problem prohibits real-time implementation and is overcome by a slicing step, which preserves the perfect reconstruction property. However, blocking effects and time-aliasing may be observed if the coefficients are modified in applications such as de-noising or transposition and time-shift of certain signal components. While slicing the signal naturally introduces a trade-off between delay and finest possible frequency resolution, the parameters can be chosen to suppress blocking artifacts and to leave the constant-Q coefficient structure intact.

The rest of this paper is organized as follows. In Section II we introduce the concepts of frames as overcomplete, stable spanning sets, with a focus on nonstationary Gabor (NSG) systems and their properties. We recall the conditions for these systems to constitute so-called *painless* frames, a special case that allows for straightforward inversion. Section III describes the construction of the CQ-NSGT by NSG frames with adaptivity in the frequency domain. This is the starting point for the sliCQ transform, which is explored in Section IV. After giving the general idea, we describe interpretation of the sliCQ-coefficients in relation to the full-length transform in Section IV-C. Subsequently, Section V is concerned with an analysis of the transforms' numerical properties, in particular computation time and complexity, as well as the quality of approximation of the CQ-NSGT coefficients by the sliCQ, accompanied by a set of simulations. Finally, in Section VI the CQ-NSGT is applied and evaluated in the analysis and processing of real-life signals. The paper is closed by a short summary and conclusion.

## II. Nonstationary Gabor Frames

Frames, first mentioned in [8], also cf. [4], [15], generalize (orthonormal) bases and allow for redundancy and thus design flexibility in signal representations. Frames may be tailored to a specific application or certain requirements such as a constant-Q frequency resolution. Loosely speaking, we wish to represent a given signal of interest as a sum of the frame members, or *atoms*, $\varphi_{n,k}$, weighted by coefficients $c_{n,k}$:

$$f = \sum_{n,k} c_{n,k}\varphi_{n,k}. \tag{1}$$

The double indexes $(n,k)$ allude to the fact that each atom has a certain location and concentration in time and frequency. Frame theory establishes conditions under which an expansion of the form (1) can be obtained with coefficients leading to stable, perfect reconstruction.

For this contribution, we only consider frames for $\mathbb{C}^L$, i.e. vector spaces of finite, discrete signals, understood as functions $f, g$ on $\mathbb{C}^L$. We denote by $\langle f, g\rangle$ the inner product of $f$ and $g$, i.e. $\langle f, g\rangle = \sum_{l=0}^{L-1} f[l]\overline{g[l]}$ and $\|f\|_2 = \sqrt{\langle f, f\rangle}$. The structures introduced here can easily be extended to the Hilbert space of quadratically integrable functions, $L^2(\mathbb{R})$.

### A. Frames

Consider a collection of atoms $\varphi_{n,k} \in \mathbb{C}^L$ with $(n,k) \in I_N \times I_K$ for finite index sets $I_N, I_K$. We then define the frame operator $\mathbf{S}$ by

$$\mathbf{S}f = \sum_{n,k} \langle f, \varphi_{n,k}\rangle \varphi_{n,k}, \tag{2}$$

for all $f \in \mathbb{C}^L$. If the linear operator $\mathbf{S}$ is invertible on $\mathbb{C}^L$, then the set of functions $\{\varphi_{n,k}\}_{(n,k)\in I_N\times I_K}$, is a *frame*[1]. In this case, we may define a *dual frame* by

$$\widetilde{\varphi_{n,k}} = \mathbf{S}^{-1}\varphi_{n,k} \tag{3}$$

and reconstruction from the coefficients $c_{n,k} = \langle f, \varphi_{n,k}\rangle$ is straightforward:

$$\begin{aligned} f &= \mathbf{S}^{-1}\mathbf{S}f = \sum_{n,k}\langle f, \varphi_{n,k}\rangle \mathbf{S}^{-1}\varphi_{n,k} = \sum_{n,k} c_{n,k}\widetilde{\varphi_{n,k}} \\ &= \mathbf{S}\mathbf{S}^{-1}f = \sum_{n,k}\langle f, \mathbf{S}^{-1}\varphi_{n,k}\rangle \varphi_{n,k} = \sum_{n,k}\langle f, \widetilde{\varphi_{n,k}}\rangle \varphi_{n,k}. \end{aligned}$$

We next introduce a case of particular importance, the so-called *Gabor frames*, for which the elements $\varphi_{n,k}$ are obtained from a single window $\varphi$ by time- and frequency-shifts along a lattice. Let $\mathbf{T}_x$ and $\mathbf{M}_\omega$ denote a time-shift by $x$ and a frequency shift (or modulation) by $\omega$, i.e.

$$\mathbf{T}_x f[l] = f[l-x] \quad \text{and} \quad \mathbf{M}_\omega f[l] = e^{2\pi i l\cdot/L} f[l],$$

[1]Note that, if $\{\varphi_{n,k}, (n,k) \in I_N \times I_K\}$ is an orthonormal basis, then $\mathbf{S}$ is the identity operator.

where $l - x$ is considered modulo $L$. In other words, this is a circular shift operation. Furthermore, we use the normalization

$$\mathcal{F}f[j] = \mathbf{f}[j] = \frac{1}{\sqrt{L}} \sum_{l=0}^{L-1} f[l] e^{-2\pi i l \cdot j/L}$$

for the discrete Fourier transform of $f$. It follows that $\mathcal{F}(\mathbf{T}_x f) = \mathbf{M}_{-x}\hat{f}$ and $\mathcal{F}(\mathbf{M}_\omega f) = \mathbf{T}_\omega \hat{f}$.

Fixing a time-shift parameter $a$ and a frequency-shift parameter $b$, with $L/a, L/b \in \mathbb{N}$, we call the collection of atoms $\mathcal{G} = \{\varphi_{n,k} = \mathbf{M}_{kb}\mathbf{T}_{na}\varphi\}_{(n,k)\in I_N \times I_K}$, with $I_N \times I_K = \mathbb{Z}_{L/a} \times \mathbb{Z}_{L/b}$, a *Gabor system*. If $\mathcal{G}$ is a frame, it is called a Gabor frame. For Gabor frames, the frame coefficients are given by samples of the short-time Fourier transform of $f$ with respect to the window $\varphi$:

$$\begin{aligned} c_{n,k} = \langle f, \varphi_{n,k}\rangle &= \langle f, \mathbf{M}_{kb}\mathbf{T}_{na}\varphi\rangle \\ &= \sum_{l=0}^{L-1} f[l]\overline{\varphi[l-na]} e^{-2\pi i l \cdot kb/L}. \end{aligned} \quad (4)$$

In a general setting, the inversion of the operator $\mathbf{S}$ poses a problem in numerical realization of frame analysis. However, for Gabor frames, it was shown in [6], that under certain conditions, usually fulfilled in practical applications, $\mathbf{S}$ is diagonal, and a dual frame can be calculated easily. This situation of *painless non-orthogonal expansions* can now be generalized to allow for adaptive resolution.

### *B. Frequency-Adaptive Painless Nonstationary Gabor Frames*

In classical Gabor frames, we obtain all samples of the STFT in (4) by applying the same window $\varphi$, shifted along a regular set of sampling points and taking an FFT of the same length. In order to achieve adaptivity of the resolution in either time or frequency, we relax the regularity of classical Gabor frames to derive *nonstationary Gabor frames*.

The original motivation for the introduction of NSGT was the desire to adapt both window size and sampling density in time, cf. [1], [11], in order to accurately resolve transient signal components. Here, we apply the same idea in frequency, i.e. adapt both the bandwidth and sampling density in frequency. From an algorithmic point of view, we apply a nonstationary Gabor system to the Fourier transform of the input signal.

The windows are constructed directly in the frequency domain by taking real-valued filters $g_k$ centered at $\omega_k$. The inverse Fourier transforms $\check{g}_k := \mathcal{F}^{-1}g_k$ are the time-reverse impulse responses of the corresponding (frequency-adaptive) filters. Therefore, we let $\check{g}_k$, $k \in I_K$, denote the members of a finite collection of band-limited windows, well-localized in time, whose Fourier transforms $g_k = \mathcal{F}\check{g}_k$ are centered around possibly irregularly (or, e.g. geometrically) spaced frequency points $\omega_k$.

Then, we select frequency dependent time-shift parameters (hop-sizes) $a_k$ as follows: if the *support* (the interval where the vector is nonzero) of $g_k$ is contained in an interval of length $L_k$, then $a_k$ is chosen such that

$$a_k \le \frac{L}{L_k} \quad \text{for all } k. \quad (5)$$

In other words, the time-sampling points have to be chosen dense enough to guarantee (5). If we denote by $g_{n,k}$ the modulation of $g_k$ by $-na_k$, i.e. $g_{n,k} = \mathbf{M}_{-na_k}g_k$, then we obtain the frame members $\varphi_{n,k}$ by setting

$$\varphi_{n,k} = \check{g_{n,k}} = \mathcal{F}^{-1}(\mathbf{M}_{-na_k}g_k) = \mathbf{T}_{na_k}\check{g}_k,$$

where $k \in I_K$ and $n = 0, \ldots, L/a_k - 1$. The system $\mathcal{G}(\mathbf{g}, \mathbf{a}) := \{\check{g_{n,k}} = \mathbf{T}_{na_k}\check{g}_k\}_{n,k}$ is a *painless nonstationary Gabor system*, as described in [1], for $\mathbb{C}^L$. We also define $\mathbf{g} := \{g_k \in \mathbb{C}^L\}_{k\in I_K}$ and $\mathbf{a} := \{a_k\}_{k\in I_K}$. By Parseval's formula, we see that the frame coefficients can be written as

$$c_{n,k} = \langle f, \check{g_{n,k}}\rangle = \langle \mathbf{f}, \mathbf{M}_{-na_k}g_k\rangle. \quad (6)$$

For convenience, we use the notation $c := \{c_k\}_{k\in I_K} := \{\{c_{n,k}\}_{n=0}^{L/a_k-1}\}_{k\in I_K}$ to refer to the full set of coefficients and channel coefficients, respectively. By abuse of notation, we indicate by $c \in \mathbb{C}^{L/a_k \times |I_K|}$ that $c$ is an irregular array with $|I_K|$ columns, the $k$-th column possessing $L/a_k$ entries. The NSG coefficients can be computed using the following algorithm.

**Algorithm 1** NSG analysis: $c = \mathbf{CQ-NSGT}_L(f, \mathbf{g}, \mathbf{a})$

1: **Initialize** $f, g_k$ for all $k \in I_K$
2: $f \leftarrow \mathbf{FFT}_L(f)$
3: **for** $k \in I_K, n = 0, \ldots, L/a_k - 1$ **do**
4: $\quad c_k \leftarrow \sqrt{L/a_k} \cdot \mathbf{IFFT}_{L/a_k}(f\overline{g_k})$
5: **end for**

Here $(\mathbf{I})\mathbf{FFT}_N$ denotes a (inverse) Fast Fourier transform of length $N$, including the necessary zero-padding preprocessing to convert the input vector to the correct length $N$. The analysis algorithm above is complemented by Algorithm 2, an equally simple synthesis algorithm that synthesizes a signal $\tilde{f}$ from a set of coefficients $c$.

**Algorithm 2** NSG synthesis: $\tilde{f} = \mathbf{iCQ-NSGT}_L(c, \tilde{\mathbf{g}}, \mathbf{a})$

1: **Initialize** $c_{n,k}, \widetilde{g_k}$ for all $n = 0, \ldots, L/a_k - 1, k \in I_K$
2: **for** $k \in I_k$ **do**
3: $\quad f_k \leftarrow \sqrt{a_k/L} \cdot \mathbf{FFT}_{L/a_k}(c_k)$
4: **end for**
5: $\tilde{f} \leftarrow \sum_{k\in I_K} f_k \widetilde{g_k}$
6: $\tilde{f} \leftarrow \mathbf{IFFT}_L(\tilde{f})$

*Remark 1:* The algorithms proposed in this section can also be applied for $a_k > L/L_k$. However, in this case, applying $(\mathbf{I})\mathbf{FFT}_{L/a_k}$ may require periodization or periodic extension, respectively, to convert the input to length $L/a_k$ or the output to length $L_k$.

If $\mathcal{G}(\mathbf{g}, \mathbf{a})$ and $\mathcal{G}(\tilde{\mathbf{g}}, \mathbf{a})$ are a pair of dual frames, then we can reconstruct a function perfectly from its NSG analysis coefficients. For more details and a proof of the following propositions, see Appendix A.

*Proposition 1:* Let $\mathcal{G}(\mathbf{g}, \mathbf{a}) = \{\check{g_{n,k}} = \mathbf{T}_{na_k}\check{g}_k\}_{n,k}$ and $\mathcal{G}(\tilde{\mathbf{g}}, \mathbf{a}) = \{\widetilde{\check{g}_{n,k}} = \mathbf{T}_{na_k}\widetilde{\check{g}_k}\}_{n,k}$ be a pair of dual frames. If $c$ is

the output of $\mathbf{CQ-NSGT}_L(f,\mathbf{g},\mathbf{a})$ (Algorithm 1), then the output $\tilde{f}$ of $\mathbf{iCQ-NSGT}_L(c,\tilde{\mathbf{g}},\mathbf{a})$ (Algorithm 2) equals $f$, i.e.

$$\tilde{f}=f,\quad \text{for all } f\in\mathbb{C}^L. \tag{7}$$

The remaining problem is to ascertain that $\mathcal{G}(\mathbf{g},\mathbf{a})$ is a frame and to compute the dual frame. The following proposition is a discrete version of an equivalent result for NSG systems in $L^2(\mathbb{R})$ and achieves both, using the painless case condition (5).

*Proposition 2:* Let $\mathcal{G}(\mathbf{g},\mathbf{a})$ an NSG system satisfying (5). This system is a frame if and only if

$$0<\sum_{k\in I_K}\frac{L}{a_k}\left|g_k[j]\right|^2<\infty,\quad \text{for all } j=0,\ldots,L-1 \tag{8}$$

and the filters generating the canonical dual frame $\mathcal{G}(\tilde{\mathbf{g}},\mathbf{a})$ are given by

$$\widetilde{g_k}[j]=\frac{g_k[j]}{\sum_{l\in I_K}\frac{L}{a_l}\left|g_l[j]\right|^2}. \tag{9}$$

In the next section, we construct a constant-Q NSG system satisfying (5) and (8).

*Remark 2:* The maximum and minimum of the sum in (8) give the upper and lower frame bound, respectively.

*Remark 3:* Note that NSG frames can be equivalently used to design general nonuniform filter banks [14], [16] in a similar manner.

## III. The CQ-NSGT Parameters: Windows and Lattices

The parameters of the NSGT can be designed as to implement various frequency-adaptive transforms. Here, we focus on the parameters leading to an NSGT with constant-Q frequency resolution, suitable for the analysis and processing of music signals, as discussed in the introduction. In constant-Q analysis, the functions $g_k$ are considered to be filters with support of length $L_k\leq L$ centered at frequency $\omega_k$ (in samples), such that for the bins corresponding to a certain frequency range, the respective center frequencies and lengths have (approximately) the same ratio. Using these filters, the CQ-NSGT coefficients $c_{n,k}$ are obtained via Algorithm 1, where $k$ indexes the frequency bins, and $n=0,\ldots,L/a_k-1$.

As detailed in [21], the construction of the filters for the CQ-NSGT depends on the following parameters: minimum and maximum frequencies $\xi_{\min}$ and $\xi_{\max}$ (in Hz), respectively, the sampling rate $\xi_s$, and the number of bins per octave $B$. The center frequencies $\xi_k$ satisfy $\xi_k=\xi_{\min}2^{(k-1)/B}$, similar to the classical CQT in [2], for $k=1,\ldots,K$, where $K$ is an integer such that $\xi_{\max}\leq\xi_K<\xi_s/2$, the Nyquist frequency. Note that the correspondence between $\xi_k$ and $\omega_k$ is the conversion ratio from Hz to samples, as detailed in the next paragraphs.

The bandwidths are set to be $\Omega_k=\xi_{k+1}-\xi_{k-1}$, for $k=2,\ldots,K-1$, which lead to a constant Q-factor $Q=\xi_k/\Omega_k=\left(2^{1/B}-2^{-1/B}\right)^{-1}$, while $\Omega_1$ and $\Omega_K$ are taken to be $\xi_1/Q$ and $\xi_K/Q$, respectively. Since the signals are real-valued, additional filters are considered which are positioned in a symmetric manner with respect to the Nyquist frequency. Moreover, to ensure that the union of filter supports cover the entire frequency axis, filters with center frequencies corresponding to the zero frequency and the Nyquist frequency are included. The values for $\xi_k$ and $\Omega_k$ over all frequency bins are summarized in Table I.

TABLE I
Center Frequency and Bandwidth Values

| $k$ | $\xi_k$ | $\Omega_k$ |
|---|---|---|
| $0$ | $0$ | $2\xi_{\min}$ |
| $1,\ldots,K$ | $\xi_{\min}2^{\frac{k-1}{B}}$ | $\xi_k/Q$ |
| $K+1$ | $\xi_s/2$ | $\xi_s-2\xi_K$ |
| $K+2,\ldots,2K+1$ | $\xi_s-\xi_{2K+2-k}$ | $\xi_{2K+2-k}/Q$ |

With these center frequencies and bandwidths, the filters $g_k$ are set to be $g_k[j]=H((j\xi_s/L-\xi_k)/\Omega_k)$, for $k=1,\ldots,K,K+2,\ldots,2K+1$, where $H$ is some continuous function centered at 0, positive inside and zero outside of $]-1/2,\ 1/2[$, i.e. each $g_k$ is a sampled version of a translated and dilated $H$. Meanwhile, $g_0$ and $g_{K+1}$ are taken to be plateau functions, i.e. continuous, compactly supported functions that are constant 1 on some interval, centered at the zero and the Nyquist frequencies respectively. Thus, each filter $g_k$ is centered at $\omega_k=\xi_kL/\xi_s$ and has support $L_k=\Omega_kL/\xi_s$.

With the aforementioned parameters, we compute the *phase-locked* CQ-NSGT coefficients as

$$c_{n,k}=\sum_{l=0}^{L-1}\mathbf{f}[l]\overline{g_k[l]}e^{2\pi i(l-\omega_k)\cdot na_k/L}.$$

This phaselock convention, while slightly different from the definition (6) above, does not affect the frame property, yet implementation is more straightforward.

It is easy to see that this choice of $\mathcal{G}(\mathbf{g},\mathbf{a})$ satisfies the conditions of Proposition 2 for any sequence $\mathbf{a}$ with $L/a_k\geq L_k$ for all $k\in I_K=\{0,\ldots,2K+1\}$. Note that while $a_k$ might be rational, $L/a_k$ must be integer-valued. Consequently, perfect reconstruction of the signal is obtained from the coefficients $c_{n,k}$ by applying Algorithm 2 with a dual frame, e.g. the canonical dual given by (9).

## IV. Real-Time Processing and the sliCQ

The CQ-NSGT implementation introduced in the previous sections a priori relies on a Fourier transform of the entire signal. This contradicts the idea of real-time applications, which require bounded delay in processing incoming samples and linear over-all complexity. These requirements can be satisfied by applying the CQ-NSGT in a blockwise manner, i.e. to (fixed length) slices of the input signal. However, the slicing process involves two important challenges: First, the windows $h_m$ used for cutting the signal must be smooth and zero-padding has to be applied to suppress time aliasing and blocking artifacts when coefficient-modification occurs. Second, the coefficients issued from the block-wise transform

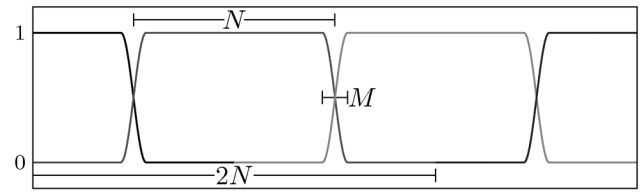



Fig. 2. Tukey windows used in the slicing process. Note that the chosen amount of zero-padding leads to a half-overlap situation.

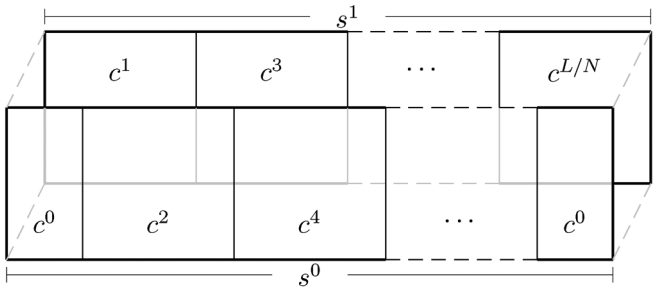



Fig. 3. Structure of the sliCQ coefficients—schematic illustration.

should be equivalent to the CQ-coefficients obtained from a full-length CQ-NSGT. This can be achieved to high precision by careful choice of both the slicing windows $h_m$ and the analysis windows $g_k$ used in the CQ-NSGT.

### *A. Structure of the sliCQ Transform*

We now summarize the individual steps of the sliCQ algorithm and introduce the involved parameters.

I) Sliced constant-Q NSGT analysis:
   1) Cut the signal $f \in \mathbb{C}^L$ into overlapping slices $f^m$ of length $2N$ by multiplication with uniform translates of a slicing window $h_0$, centered at 0.
   2) For each $f^m$, obtain coefficients $c^m \in \mathbb{C}^{2N/a_k \times |I_K|}$, by applying $\mathbf{CQ-NSGT}_{2N}(f, \mathbf{g}, \mathbf{a})$ (Algorithm 1).
   3) Due to the overlap of the slicing windows, cf. Fig. 2, each time index is related to two consecutive slices. For visualization and processing, the slice coefficients $c^m$ are re-arranged into a 2-layer array $s$, with $s := \{s^l\}_{l\in\{0,1\}} \in \mathbb{C}^{2\times L/a_k \times |I_K|}$, cf. Fig. 3.

II) Sliced constant-Q NSGT synthesis:
   1) Retrieve $c^m$ by partitioning $s$.
   2) Compute the dual frame $\mathcal{G}(\tilde{\mathbf{g}}, \mathbf{a})$ for $\mathcal{G}(\mathbf{g}, \mathbf{a})$ and, for all $m$, $\tilde{f}^m = \mathbf{iCQ-NSGT}_{2N}(c^m, \tilde{\mathbf{g}}, \mathbf{a})$ (Algorithm 2).
   3) Recover $f$ by (windowed) overlap-add.

Note that $L$ must be a multiple of $2N$; this is achieved by zero-padding, if necessary. By construction, the positions $(n, k)$ of the coefficients in $s^l$ reflect their time-frequency position with respect to the full-length signal, for $l = 0, 1$.

### *B. Computation of a Sliced Constant-Q NSGT*

The *sliced constant-Q NSGT* (sliCQ) coefficients of $f$ with respect to $h_0$ and $\mathcal{G}(\mathbf{g}, \mathbf{a})$ and slice length $2N$ are obtained according to Algorithm 3.

**Algorithm 3** sliCQ analysis: $s = \mathbf{sliCQ}_{L,N}(f, h_0, \mathbf{g}, \mathbf{a})$

**Initialize** $f, h_0, g_k$ for all $k \in I_K$
$m \leftarrow 0$
**for** $m = 0, \ldots L/N - 1$ **do**
  **for** $j = 0, \ldots 2N - 1$ **do**
    $f^m[j] \leftarrow f\mathbf{T}_{mN}h_0[j + (m-1)N]$
  **end for**
  $c^m \leftarrow \mathbf{CQ-NSGT}_{2N}(f, \mathbf{g}, \mathbf{a})$
  $l \leftarrow (m \bmod 2)$
  **for** $k \in I_K, n^s = 0, \ldots, 2N/a_k - 1$ **do**
    $s^l_{n^s+(m-1)N/a_k,k} \leftarrow c^m_{n^s,k}$
  **end for**
**end for**

Note that in this and the following algorithm, negative indices are used in a circular sense, with respect to the maximum admissible index, e.g. $f[-j] := f[L - j]$ or $s^l_{-n,k} := s^l_{L/a_k-n,k}$. As the CQ-NSGT analysis before, Algorithm 3 is complemented by a synthesis algorithm with similar structure, Algorithm 4, that synthesizes a signal $\tilde{f}$ from a 2-layer coefficient array $s$.

**Algorithm 4** sliCQ synthesis: $\tilde{f} = \mathbf{isliCQ}_{L,N}(s, \tilde{h}_0, \tilde{\mathbf{g}}, \mathbf{a})$

**Initialize** $s, \tilde{h}_0, \tilde{g}_k$ for all $k \in I_K$
$m \leftarrow 0$
$\tilde{f} \leftarrow \mathbf{0}_L$
**for** $m = 0, \ldots L/N - 1$ **do**
  $l \leftarrow (m \bmod 2)$
  **for** $k \in I_K, n^s = 0, \ldots, 2N/a_k - 1$ **do**
    $c^m_{n^s,k} \leftarrow s^l_{n^s+(m-1)N/a_k,k}$
  **end for**
  $\tilde{f}^m \leftarrow \mathbf{iCQ-NSGT}_{2N}(c^m, \tilde{\mathbf{g}}, \mathbf{a})$
  **for** $j = 0, \ldots 2N - 1$ **do**
    $\tilde{f}[j+(m-1)N] \leftarrow \tilde{f}[j+(m-1)N] + \tilde{f}^m[j]\tilde{h}_0[j-N]$
  **end for**
**end for**

The following proposition states that $f$ is perfectly recovered from its sliCQ coefficients by applying Algorithm 4, see Appendix B for a proof.

*Proposition 3:* Let $\mathcal{G}(\mathbf{g}, \mathbf{a})$ and $\mathcal{G}(\tilde{\mathbf{g}}, \mathbf{a})$ be dual NSG systems for $\mathbb{C}^{2N}$. Further let $h_0, \tilde{h}_0 \in \mathbb{C}^L$ satisfy

$$\sum_{m=0}^{L/N-1} \mathbf{T}_{mN}\left(h_0\overline{\tilde{h}_0}\right) \equiv 1. \tag{10}$$

If $s$ is the output of $\mathbf{sliCQ}_{L,N}(f, h_0, \mathbf{g}, \mathbf{a})$ (Algorithm 3), then the output $\tilde{f}$ of $\mathbf{isliCQ}_{L,N}(s, \tilde{h}_0, \tilde{\mathbf{g}}, \mathbf{a})$ (Algorithm 4) equals $f$, i.e., $\tilde{f} = f$.

### C. The Relation Between CQ-NSGT and sliCQ

To maintain perfect reconstruction in the final overlap-add step in Algorithm 4, we assume

$$h_m = \mathbf{T}_{mN} h_0 \text{ with } \sum_{m=0}^{L/N-1} h_m \equiv 1, \tag{11}$$

and use a dual window $\tilde{h}_0$ satisfying (10) in the synthesis process.

Another obvious option for the design of the slicing windows is to require $\sum_m h_m^2 \equiv 1$, which would allow for using the same windows in the final overlap-add step. However, if we want to approximate the true CQ-coefficients as obtained from a full-length transform, (11) is the more favorable condition.

In our implementation, *slicing* of the signal is accomplished by a uniform partition of unity constructed from a Tukey window $h_0$ with essential length $N$ and transition areas of length $M$, for some $N, M \in \mathbb{N}$ with $M < N$ (usually $M \ll N$). The slicing windows are symmetrically zero-padded to length $2N$, reducing time aliasing significantly. The uniform partition condition (11) leads to close approximation of the full-length CQ-NSGT by sliCQ. This correspondence between the sliCQ and the corresponding full-length CQ-NSGT is made explicit in the following proposition, proven in Appendix B.

*Proposition 4:* Let $\mathcal{G}(\mathbf{g}^{\mathcal{L}}, \mathbf{a})$ be a nonstationary Gabor system for $\mathbb{C}^L$. Further, let $h_0 \in \mathbb{C}^L$ be such that (11) holds and define $g_k \in \mathbb{C}^{2N}$, for all $k \in I_K$ by

$$g_k[j] = g_k^{\mathcal{L}}\left[\frac{jL}{(2N)}\right].$$

For $f \in \mathbb{C}^L$, denote by $c \in \mathbb{C}^{L/a_k \times |I_K|}$ the CQ-NSGT coefficients of $f$ with respect to $\mathcal{G}(\mathbf{g}^{\mathcal{L}}, \mathbf{a})$ and by $s \in \mathbb{C}^{2\times L/a_k \times |I_K|}$ the sliCQ coefficients of $f$ with respect to $h_0$ and $\mathcal{G}(\mathbf{g}, \mathbf{a})$. Then

$$\left|s_{n,k}^0 + s_{n,k}^1 - c_{n,k}\right| \le \|f\|_2 \left( \left\| (1 - h_0 - h_1)\mathbf{T}_{n^s a_k} \check{g}_k^{\mathcal{L}} \right\|_2 + \left\| (h_0 + h_1) \sum_{j=1}^{\frac{L}{2N}-1} \mathbf{T}_{n^s a_k + 2jN} \check{g}_k^{\mathcal{L}} \right\|_2 \right) \tag{12}$$

for $n = mN/a_k + n^s$, with $m = 0, \ldots, L/N - 1$ and $n^s = 0, \ldots, N/a_k - 1$.

*Remark 4:* In practice, $\check{g}_k^{\mathcal{L}}$ is chosen such that the translates $\mathbf{T}_{na_k}\check{g}_k^{\mathcal{L}}$ are *essentially concentrated* in

$$I_{N,M} = \left[-\frac{N-M}{2}, N + \frac{N-M}{2}\right],$$

i.e. $\|\mathbf{T}_{na_k}\check{g}_k^{\mathcal{L}}\chi_{\mathbb{R}\setminus I_{N,M}}\|_2 \ll \|\mathbf{T}_{na_k}\check{g}_k^{\mathcal{L}}\|_2$, for all $n = 0, \ldots, N/a_k - 1$. Therefore, the value of (12) is negligibly small. While more precise estimates of the error are beyond the scope of the present contribution, numerical evaluation of the approximation quality is given in Section V-C.

As a consequence of the previous proposition, we define the *sliCQ spectrogram* as $|s^0 + s^1|^2$ and propose to simultaneously treat $s_{n,k}^0$ and $s_{n,k}^1$, corresponding to the same time-frequency position, when processing the coefficients.

## V. Numerical Analysis and Simulations

In this section we treat the computational complexity of CQ-NSGT and sliCQ and how they compare to one another. In [21] it was shown that despite superlinear complexity, CQ-NSGT outperforms state-of-the-art implementations of the classical constant-Q transform. Since sliCQ is a linear cost algorithm, it further improves the efficiency of the CQ-NSGT for sufficiently long signals. Section V-C provides experimental results confirming the good approximation of CQ-NSGT by the corresponding sliCQ coefficients, cf. Proposition 4.

The CQ-NSGT and sliCQ Toolbox (for MATLAB and Python) used in this contribution is available at http://www.univie.ac.at/nonstatgab/slicq, alongside extended experimental results complementing those presented in Section VI.

### A. Computation Time and Computational Complexity

We assume the number of filters $|I_K|$ in the CQ-NSGT to be independent of the signal length $L$ and Proposition 2 to hold, in particular $L/a_k \ge L_k$. The support size $L_k$ of each filter $g_k$ depends on $L$. Hence, the number of operations for Algorithm 1 is as follows:

$$\mathcal{O}\left( \underbrace{L\log(L)}_{\mathrm{FFT}_L} + \sum_{k\in I_K} \underbrace{L/a_k \log\left(\frac{L}{a_k}\right)}_{\mathrm{IFFT}_{\frac{L}{a_k}}} + \underbrace{L_k}_{f\cdot\overline{g_k}} \right).$$

With $L_k$ and $L/a_k$ bounded by $L$, this can be simplified to $\mathcal{O}(L\log(L))$.

The computation of the dual frame involves inversion of the multiplication operator $\mathbf{S}$ and applying the resulting operator $\mathbf{S}^{-1}$ to each filter. This results in $\mathcal{O}(2\sum_{k\in I_K} L_k) = \mathcal{O}(L)$ operations, where the support of the $g_k$ was taken into account.

Complexity of Algorithm 2 can be derived to be $\mathcal{O}(L\log(L))$, analogous to Algorithm 1.

For $\mathbf{sliCQ}_{L,N}$ (Algorithm 3), we assume the slice length $2N$ to be independent of $L$, resulting in a computational complexity of

$$\mathcal{O}\left( \underbrace{\frac{L}{N}}_{\#\mathrm{slices}} \cdot \left( \underbrace{2N\log(2N)}_{\mathbf{CQ-NSGT}_{2N}} + \underbrace{2N}_{f\cdot\mathbf{T}_{mN}\overline{h_0}} \right) \right) = \mathcal{O}(L).$$

Both the dual frame and $\tilde{h}_0$ can be precomputed independent of $L$, whilst Algorithm 4 is of complexity $\mathcal{O}(L)$, analogous to Algorithm 3.

### B. Performance Evaluation

A comparison of the CQ-NSGT algorithm with previous constant-Q implementations was given in [21]. Fig. 4 reproduces and extends some of the results; it shows, for both the constant-Q implementation provided in [18] and CQ-NSGT, mean computation duration and variance for analysis followed by reconstruction, against signal length. The plot also illustrates the dependence of CQ-NSGT on the prime factor decomposition of the signal length $L$.

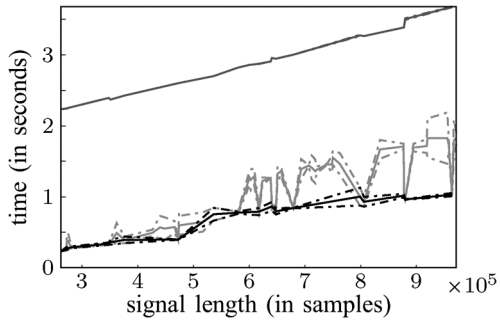



Fig. 4. Computation time versus signal length of the CQ transform (dark gray) and CQ-NSGT. For the CQ-NSGT we show separate graphs including (light gray), respectively neglecting prime signal lengths (black). Graphs show the mean performance (solid) and variance (dashed) over 50 iterations.

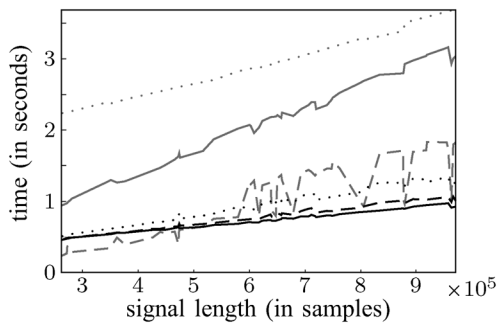



Fig. 5. Computation time versus signal length of the CQ transform (dotted gray), CQ-NSGT (dashed gray) and various sliCQ transforms. The sliCQ transforms were taken with slice lengths 4096 (solid gray), 16384 (dotted black), 32768 (dashed black) and 65536 (solid black) samples.

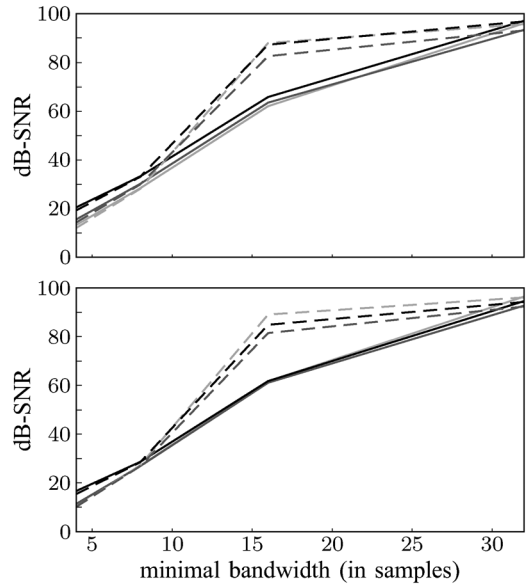



Fig. 6. SliCQ coefficient approximation error against the minimal admissible bandwidth for Set 1 (top) and Set 2 (bottom). All transforms use Blackman-Harris windows in the CQ-NSGT step. Solid and dashed lines represent long (1/4 slice length) and short (1/128 slice length) transition areas respectively, while colors correspond to the slice length: 4096 (light gray), 16384 (dark gray) and 65536 samples (black).

Fig. 5 illustrates the performance of sliCQ compared to the constant-Q and CQ-NSGT algorithms shown in Fig. 4. Linearity of the sliCQ algorithm becomes evident, with deviations occurring due to unfavorable FFT lengths $2N/a_k$ in $(\mathbf{i})\mathbf{CQ}-\mathbf{NSGT}_{2N}$. Performance improvements for increasing slice length can be attributed to the advanced nature of MATLAB's internal FFT algorithm, as compared to the current implementation of the sliCQ framework.

The performance of the involved algorithms does not depend on signal content. Consequently, random signals were used in the performance experiments, although we implicitly assumed the signals to be sampled at 44.1 kHz. All the results represent transforms with 48 bins per octave, minimum frequency 50 Hz and maximum frequency 22 kHz, in Section VI a maximum frequency of 20 kHz is used instead. For a more comprehensive comparison of the CQ-NSGT to previous constant-Q transforms, please refer to [21]. Results for other parameter values do not differ drastically and are omitted.

All computation time experiments were run in MATLAB R2011a on a 3 Gigahertz Intel Core 2 Duo machine with 2 Gigabytes of RAM running Kubuntu 10.04 using the MATLAB toolboxes available at http://www.elec.qmul.ac.uk/people/anssik/cqt/ and http://www.univie.ac.at/nonstatgab/.

### C. Approximation Properties

To verify the approximate equivalence of the sliCQ coefficients to those of a full-length CQ-NSGT and thus to a constant-Q transform, we computed the norm difference between $s^0+s^1$ and $c$ as in Proposition 4, for two sets of fundamentally different signals. Set 1 contains 50 random, complex-valued signals of $2^{20}$ samples length, while Set 2 consists of 90 music samples of the same length, sampled at 44.1 kHz each, covering pop, rock, jazz and classical genres. The signals of the second set are well-structured and often well-concentrated in the time-frequency plane, characteristics that the first set lacks completely.

For discretization reasons as well as to achieve good concentration of $g_k^{\mathcal{L}}$ in Proposition 4, sliCQ implementations must impose a lower bound on the length of $g_k$. Approximation results for various lower bounds on the filter length are summarized in Fig. 6, showing the mean approximation quality over the whole set.

All errors are given in signal-to-noise ratio, scaled in dB:

$$20\log_{10}\frac{\|c\|_2}{\|c-(s^0+s^1)\|_2}$$

Fig. 6 shows that, independent of other parameters, a minimal filter length smaller than 8 samples leads to a representation that is visibly different from, while values above 16 samples yield coefficients that are largely equivalent to those of a constant-Q transform. We can see that the slice length itself has rather small influence on the results, while the interplay of slicing window shape, specified by the ratio of transition area length to slice length, and minimal filter length is illustrated nicely; remarkably, this ratio influences the approximation quality mainly for moderately well localized filters. This is in correspondence with the characterization given in (12): the *circular overspill*, given by the second term of the right hand side in (12), depends on the shape and support of the sum of two adjacent slicing windows, in particular for moderately well localized filters. If the windows are very well localized, the overspill is small independent of the particular shape of the slicing area. On the other hand,

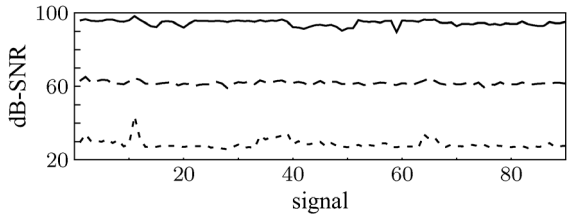



Fig. 7. Coefficient approximation error (12) for all signals from Set 2 and slice and transition length of 65536, resp. 16384 samples. Line style indicates the minimal bandwidth: 8 (dotted), 16 (dashed) and 32 (solid) samples.

very badly localized windows make the distinct influence of the slicing windows negligible. Finally, a comparison of the top and bottom graphs in Fig. 6 shows that the approximation quality is largely independent of the signal class. For Set 1 the variance is generally negligible ($<$0.1 dB) and was omitted. Despite some outliers in Set 2, we have found the approximation quality to depend on the minimal filter length in a stable way, cf. Fig. 7. These outliers can be attributed to signals particularly sparse (smaller error) or dense (larger error) in low frequency regions, where $\check{g_k^{\mathcal{L}}}$ is least concentrated.

## VI. Experiments on Applications

Experiments in [21] show how the CQ-NSGT can be applied in the processing of signals taking advantage of the logarithmic frequency scaling and the perfect reconstruction property. In particular, the transposition of a harmonic structure amounted to just a translation of the spectrum along frequency bins, while the masking of the CQ-NSGT coefficients allowed for the extraction or suppression of a component of the signal. In our experiment, we show that the two procedures can be used to modify a portion of a signal.

Fig. 8 shows masks for isolating a transient part and the corresponding sinusoidal part of a Glockenspiel signal, created using an ordinary image manipulation program. Therein, the layers paradigm has been used to be able to quickly switch on and off the masks in order to accurately adapt them to the CQ-NSGT representation of the audio. An "inverse mask" is also constructed for the remainder part of the signal, essentially decomposing the signal into transient, sinusoidal and background portions. The masks have been drawn in the logarithmic domain, to be able to handle the dynamics of the audio. They are linearly scaled in dB units, so that 0 in the mask corresponds to $10^{-5}$ ($-100$ dB) and 1 corresponds to 1 (0 dB).

While keeping the transient part, the isolated sinusoidal component of the signal is transposed upward by 2 semitones, corresponding to 8 frequency bins. The transient, the remainder, and the modified sinusoidal coefficients are then added and the inverse transform is applied to obtain the resulting processed signal. For ease of use, this process is done with a rectangular representation of the slices, obtained by choosing $L/a_k$ constant for all frequency bands which corresponds to a sinc-interpolation of the coefficients.

Fig. 9 compares the CQ-NSGT spectrograms of the original and the modified signal, while Fig. 10 shows the results for the same experiment using sliCQ transforms with different slice

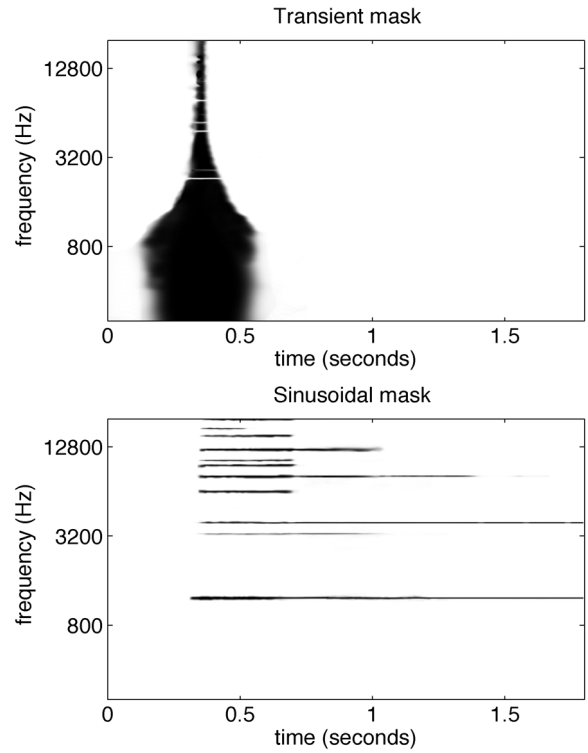



Fig. 8. Masks for extracting a transient (top) and sinusoidal component (bottom) of the Glockenspiel signal. The gray level plot describes the amplitude of the mask, with black and white representing 1 and 0, respectively.

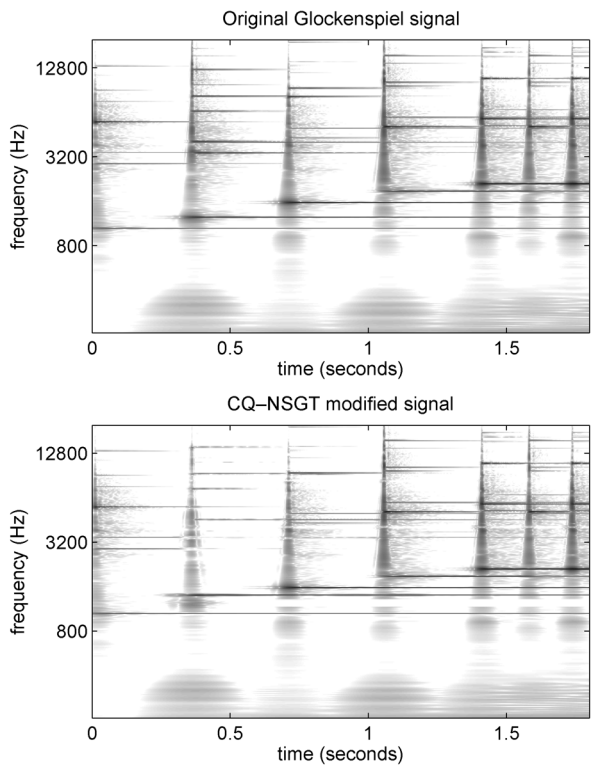



Fig. 9. CQ-NSGT spectrograms showing an excerpt of the Glockenspiel signal before (top) and after transposition of a component (bottom).

lengths. Note that the plots show the spectrogram of the synthesized signal, not the time-frequency coefficients before synthesis. Further, the exact same mask was used for CQ-NSGT and

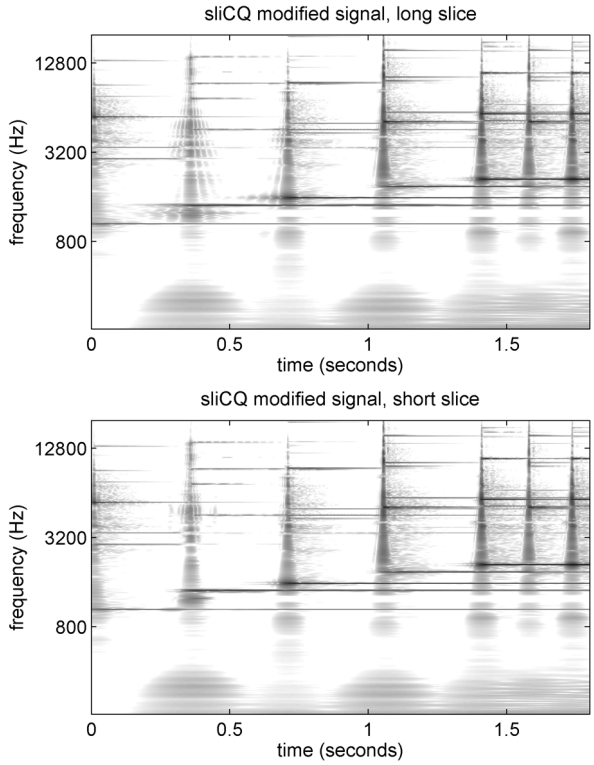



Fig. 10. sliCQ spectrograms showing an excerpt of the Glockenspiel signal after transposition of a component. The top plot was done with a slice length of 50000 and a transition area of 20000 samples, the bottom plot with a slice length of 5000 and a transition area of 2000 samples.

sliCQ transpositions. The sound files for this and other transposition experiments are available at http://www.univie.ac.at/nonstatgab/slicq. A script for the Python toolbox that executes the experiment, is available on the same page.

For synthesis, performed from modified coefficients, as opposed to mere reconstruction, an evaluation of the results is a highly non-trivial matter. This is due to the lack of a properly defined notion of *accuracy* or the existence of a *target signal*, not only for the algorithms presented here, but for any analysis/synthesis based signal processing framework. Thus, while the examples in this section should indicate that CQ-NSGT synthesis and sliCQ synthesis can produce results in accordance with intuition, an in-depth treatment of this subject is far beyond the scope of this article.

## VII. Summary and Conclusion

In this contribution, we have introduced a framework for real-time implementation of an invertible constant-Q transform based on frame theory. The proposed framework allows for straight-forward generalization to other non-linear frequency scales, such as mel- or Bark scale, cp. [9]. While real-time processing is possible by means of a preprocessing step, we investigated the possible occurrence of time-aliasing. We provided a numerical evaluation of computation time and quality of approximation of the true NSGT coefficients.

In analogy to the classical phase vocoder, phase issues have to be addressed, if CQ-transformed coefficients are processed, cp. [12], [13], [17]. While preliminary experiments using the proposed framework for real-life signals were presented, undesired phasing effects, mainly due to the contribution of a signal component to several adjacent filters, will be investigated in detail in future work. Furthermore, future work will consider the efficient realization of adaptivity in both time and frequency by varying the length of the preprocessing windows used for slicing.

## Appendix

### A. Derivation of CQ-NSGT Properties

*Proof of Proposition 1:* By Algorithm 1, we have

$$\begin{aligned} c_{n,k} &= c_k[n] \\ &= \sqrt{\frac{L}{a_k}} \frac{1}{\sqrt{\frac{L}{a_k}}} \sum_{m=0}^{L/a_k-1} \sum_{l=0}^{a_k-1} (\mathbf{f}\overline{g_k}) \left[m + l\frac{L}{a_k}\right] e^{\frac{2\pi i n m a_k}{L}} \\ &= \sum_{m=0}^{L/a_k-1} \sum_{l=0}^{a_k-1} (\mathbf{f}\mathbf{M}_{na_k}\overline{g_k}) \left[m + l\frac{L}{a_k}\right] \end{aligned} \tag{13}$$

Since $L/a_k \geq L_k$, only one element of the inner sum above is non-zero, for each $m \in \{0, \ldots, L/a_k - 1\}$. It follows that

$$c_{n,k} = \langle \mathbf{f}, \mathbf{M}_{-na_k} g_k \rangle. \tag{14}$$

Inserting into Algorithm 2 yields, for all $j \in \{0, \ldots, L-1\}$,

$$\begin{aligned} \hat{\tilde{f}}[j] &= \sum_{k \in I_K} \sum_{n=0}^{L/a_k-1} c_{n,k} e^{-2\pi i n m a_k / L} \tilde{g_k}[j] \\ &= \sum_{k \in I_K} \sum_{n=0}^{L/a_k-1} \langle \mathbf{f}, \mathbf{M}_{-na_k} g_k \rangle \mathbf{M}_{-na_k} \tilde{g_k}[j], \end{aligned}$$

the discrete frame synthesis formula. By assumption, $\mathcal{G}(\mathbf{g}, \mathbf{a})$ and $\mathcal{G}(\tilde{\mathbf{g}}, \mathbf{a})$ are dual NSG frames and thus

$$\hat{\tilde{f}}[j] = \mathbf{f}[j], \quad \text{for all } j \in \{0, \ldots, L-1\}.$$

Applying the inverse discrete Fourier transform completes the proof. ∎

*Proof of Proposition 2:* Denote by $J_k$ an interval of length $L_k$, $L_k$ as in Section II, containing the support of $g_k$. By assumption

$$0 < \sum_{k \in I_K} |g_k[j]|^2 < \infty, \quad \text{for all } j = 0, \ldots, L-1$$

and $L/a_k \geq L_k = |J_k|$. Note that the frame operator (2) can be written as follows

$$\begin{aligned} \mathbf{S}f[j] &= \sum_{k \in I_K} \sum_{n=0}^{L/a_k-1} \langle f, \mathbf{M}_{-na} g_k \rangle \mathbf{M}_{-na} g_k[j] \\ &= \sum_{k \in I_K} \sqrt{\frac{L}{a_k}} \sum_{n=0}^{L/a_k-1} \mathbf{IFFT}_{\frac{L}{a_k}}(f\overline{g_k})[n] g_k[j] e^{-2\pi i n j a_k / L} \\ &= \sum_{k \in I_K} \frac{L}{a_k} \mathbf{FFT}_{\frac{L}{a_k}} \left( \mathbf{IFFT}_{\frac{L}{a_k}}(f\overline{g_k}) \right)[j] g_k[j], \end{aligned} \tag{15}$$

for all $f \in \mathbb{C}^L$. Furthermore, with $\chi_{J_k}$ the characteristic function of the interval $J_k$,

$$\begin{aligned} f\overline{g_k} &= \chi_{J_k} \sum_{l=0}^{a_k-1} \mathbf{T}_{\frac{LL}{a_k}}(f\overline{g_k}) \\ &= \chi_{J_k} \mathbf{FFT}_{\frac{L}{a_k}} \left( \mathbf{IFFT}_{\frac{L}{a_k}}(f\overline{g_k}) \right) \end{aligned}$$

and, obviously, $g_k = \chi_{J_k} g_k$. Inserting into (15) yields

$$\begin{aligned} \mathbf{S}f[j] &= \sum_{k \in I_K} \frac{L}{a_k} (f\overline{g_k})[j] g_k[j] \\ &= f[j] \sum_{k \in I_K} \frac{L}{a_k} |g_k|^2[j]. \end{aligned} \tag{16}$$

With the sum bounded above and below, the inverse frame operator can be written as

$$\mathbf{S}^{-1} f[j] = f[j] \left( \sum_{k \in I_K} \frac{L}{a_k} |g_k|^2[j] \right)^{-1}, \text{ for all } f \in \mathbb{C}^L. \tag{17}$$

Since the elements of the canonical dual frame are given by (3), this completes the proof. ∎

### B. Derivation of sliCQ Properties

*Proof of Proposition 3:* According to Proposition 1, $\tilde{f}^m$, the output of **iCQ-NSGT** in Step 9 of Algorithm 4 satisfies to $f^m[j] = (f \cdot \mathbf{T}_{mN} h_0)[j + (m-1)N]$. Since $\sum_m \mathbf{T}_{mN}(h_0 \tilde{h}_0) \equiv 1$ holds,

$$\tilde{f} = \sum_m (f \cdot \mathbf{T}_{mN} h_0) \mathbf{T}_{mN} \overline{\tilde{h}_0} = f \cdot \sum_m \mathbf{T}_{mN}(h_0 \overline{\tilde{h}_0}) = f$$

follows. ∎

*Proof of Proposition 4:* Since $g_k$ is obtained by sampling $g_k^{\mathcal{L}}$ with sampling period $L/2N$, the (inverse) Fourier transform $\check{g}_k$ of $g_k$ is given by periodization of $g_k^{\mathcal{L}}$ as follows:

$$\check{g_k}[l] = \sum_{j=0}^{\frac{L}{2N}-1} \check{g_k^{\mathcal{L}}}[l + j \cdot 2N]. \tag{18}$$

Recall from (6) that the CQ-NSGT coefficients of $f$ with respect to $\mathcal{G}(\mathbf{g}^{\mathcal{L}}, \mathbf{a})$ are given by $c_{n,k} = \langle f, \mathbf{T}_{na_k} \check{g_k^{\mathcal{L}}} \rangle$, while the CQ-NSGT coefficients $c^m$ of $f^m$ are, for $m = 0, \ldots, L/N - 1$, $n^s = 0, \ldots, (2N/a_k) - 1$ and $k \in I_K$

$$\begin{aligned} c^m_{n^s,k} &= \langle \widehat{f^m}, g_{n^s,k} \rangle = \left\langle \widehat{f^m}, \mathbf{M}_{-n^s a_k} g_k \right\rangle \\ &= \langle f^m, \mathbf{T}_{n^s a_k} \check{g}_k \rangle \\ &= \left\langle f, h_m \sum_{j=0}^{\frac{L}{2N}-1} \mathbf{T}_{n^s a_k + (m-1+2j)N} \check{g_k^{\mathcal{L}}} \right\rangle, \end{aligned} \tag{19}$$

where the final inner product is taken over $\mathbb{C}^L$. Observe that every $n = 0, \ldots, (L/a_k) - 1$ can be written as $n = m(N/a_k) + n^s$ with $n^s$ from $0, \ldots, (N/a_k) - 1$ and thus

$$\begin{aligned} s^0_{n,k} + s^1_{n,k} &= c^m_{n^s + N/a_k, k} + c^{m+1}_{n^s,k} \\ &= \left\langle f, (h_m + h_{m+1}) \sum_{j=0}^{\frac{L}{2N}-1} \mathbf{T}_{n^s a_k + (m+2j)N} \check{g_k^{\mathcal{L}}} \right\rangle \\ &= \left\langle f, \mathbf{T}_{n^s a_k + mN} \check{g_k^{\mathcal{L}}} \right\rangle + R[n] \\ &= \left\langle f, \mathbf{T}_{a_k \left( \frac{mN}{a_k} + n^s \right)} \check{g_k^{\mathcal{L}}} \right\rangle + R[n] \\ &= c_{n,k} + R[n]. \end{aligned} \tag{20}$$

Here,

$$\begin{aligned} R[n] = &\left\langle f, (h_m + h_{m+1} - 1) \mathbf{T}_{n^s a_k + mN} \check{g_k^{\mathcal{L}}} \right\rangle \\ &+ \left\langle f, (h_m + h_{m+1}) \sum_{j=1}^{\frac{L}{2N}-1} \mathbf{T}_{n^s a_k + (m+2j)N} \check{g_k^{\mathcal{L}}} \right\rangle. \end{aligned} \tag{21}$$

Hence $s^0_{n,k} + s^1_{n,k} - c_{n,k} = R[n]$. The result follows from Cauchy-Schwartz' inequality, applied to the case $m = 0$, observing independence from $m$. ∎

## ACKNOWLEDGMENT

The authors wish to thank the reviewers for their extremely helpful and constructive remarks.

## REFERENCES

[1] P. Balazs, M. Dörfler, F. Jaillet, N. Holighaus, and G. A. Velasco, "Theory, implementation and applications of nonstationary Gabor Frames," *J. Comput. Appl. Math.*, vol. 236, no. 6, pp. 1481–1496, 2011.

[2] J. Brown, "Calculation of a constant Q spectral transform," *J. Acoust. Soc. Amer.*, vol. 89, no. 1, pp. 425–434, 1991.

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

[4] A. Chebira and J. Kovacevic, "Life beyond bases: The advent of frames (part I)," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 86–104, Jul. 2007.

[5] M. Cranitch, M. Cychowski, and D. FitzGerald, "Towards an inverse constant Q transform," in *Proc. Audio Eng. Soc. Conv.*, 2006, vol. 120, no. 5.

[6] I. Daubechies, A. Grossmann, and Y. Meyer, "Painless nonorthogonal expansions," *J. Math. Phys.*, vol. 27, no. 5, pp. 1271–1283, May 1986.

[7] M. Dolson, "The phase vocoder: A tutorial," *Comput. Mus. J.*, vol. 10, no. 4, pp. 11–27, 1986.

[8] R. J. Duffin and A. C. Schaeffer, "A class of nonharmonic Fourier series," *Trans. Amer. Math. Soc.*, vol. 72, pp. 341–366, 1952.

[9] G. Evangelista, M. Dörfler, and E. Matusiak, "Phase vocoders with arbitrary frequency band selection," in *Proc. 9th Sound and Music Comput. Conf. (SMC'12)*, Copenhagen, Denmark, Jul. 2012.

[10] H. G. Feichtinger and T. Strohmer, *Gabor Analysis and Algorithms. Theory and Applications*. Boston, MA: Birkhäuser, 1998.

[11] F. Jaillet, "Représentation et traitement temps-fréquence des signaux audionumériques pour des applications de design sonore," Ph.D. dissertation, Univ. de la Méditerranée, Aix-Marseille II, France, 2005.

[12] J. Laroche and M. Dolson, "Phase-vocoder: About this phasiness business," in *Proc. IEEE ASSP Workshop Applicat. of Signal Process. to Audio Acoust.*, Oct. 1997, p. 4.

[13] J. Laroche and M. Dolson, "Improved phase vocoder time-scale modification of audio," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 3, pp. 323–332, May 1999.

[14] J. Li, T. Nguyen, and S. Tantaratana, "A simple design method for near-perfect-reconstruction nonuniform filter banks," *IEEE Trans. Signal Process.*, vol. 45, no. 8, pp. 2105–2109, Aug. 1997.
[15] R. Marks, *Handbook of Fourier Analysis and its Applications*. Oxford, U.K.: Oxford Univ. Press, 2009.
[16] K. Nayebi, I. Barnwell, T. P. , and M. Smith, "Nonuniform filter banks: A reconstruction and design theory," *IEEE Trans. Signal Process.*, vol. 41, no. 3, pp. 1114–1127, Mar. 1993.
[17] J. Roe, *Lectures on Coarse Geometry*, ser. University Lecture Series. Providence, RI: Amer. Math. Soc., 2003, vol. 31.
[18] C. Schörkhuber and A. Klapuri, "Constant-Q toolbox for music processing," in *Proc. 7th Sound and Music Comput. Conf. (SMC'10)*, Barcelona, Spain, Jul. 2010.
[19] I. Selesnick and I. Bayram, "Frequency-domain design of overcomplete rational-dilation wavelet transforms," *IEEE Trans. Signal Process.*, vol. 57, no. 8, pp. 2957–2972, Aug. 2009.
[20] J. O. Smith, "Audio FFT filter banks," in *Proc. 12th Int. Conf. Digital Audio Effects (DAFx-09)*, Como, Italy, Sep. 2009.
[21] G. A. Velasco, N. Holighaus, M. Dörfler, and T. Grill, "Constructing an invertible constant-Q transform with non-stationary Gabor frames," in *Proc. 14th Int. Conf. Digital Audio Effects (DAFx-11)*, Paris, France, Sep. 2011.
[22] J. Youngberg and S. Boll, "Constant-Q signal analysis and synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP'78)*, 1978, vol. 3, pp. 375–378.

**Nicki Holighaus** studied mathematics and theoretical computer sciences at Justus— Liebig— University, Giessen, Germany. After graduation in 2010, he took a position as research assistant at the Numerical Harmonic Analysis Group (NuHAG), Faculty of Mathematics, University of Vienna, Austria, where he currently pursues a Ph.D. degree.

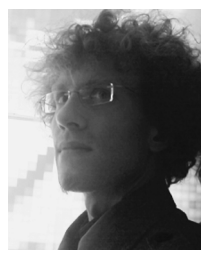
His research interests include applied frame theory and time-frequency analysis, adaptive time-frequency techniques and signal processing.

**Monika Dörfler** obtained her PhD in Mathematics from the University of Vienna and is a researcher at the Faculty of Mathematics. She studied piano at the Music University of Vienna and is working in the field of applied mathematics for audio signal processing.

She is interested in the interplay of local and global aspects of time-frequency analysis, and focuses on the benefits of theoretical results in practical applications. She is heading the interdisciplinary research project AudioMiner.

**Gino Angelo Velasco** received his B.S. and M.S. degree in Mathematics from the University of the Philippines Diliman in 2001 and 2006, respectively. He is an instructor at the Institute of Mathematics, University of the Philippines Diliman and is pursuing a Ph.D. degree at the Numerical Harmonic Analysis Group (NuHAG), Faculty of Mathematics, University of Vienna.

His research interests include approximation theory, time-frequency analysis and its application to signal processing.



**Thomas Grill** obtained his PhD in Sound and Music Computing from the University of Music and Performing Arts in Graz, Austria. He holds a lectureship for sonic art at the University of Applied Arts Vienna, Austria, and a position as researcher at the Austrian Research Institute for Artificial Intelligence (OFAI).

Also being an electroacoustic composer and performer, his research interests include auditory and cross-modal perception, sound and music computing, and new interfaces for musical expression.