PITCH SHIFTING OF AUDIO SIGNALS USING THE CONSTANT-Q TRANSFORM

Christian Schörkhuber,
Institute of Electronic Music and Acoustics
University of Music and Performing Arts, Graz
schoerkhuber@iem.at

Anssi Klapuri,
Centre for Digital Music,
Queen Mary University of London
anssi.klapuri@elec.qmul.ac.uk

Alois Sontacchi,
Institute of Electronic Music and Acoustics
University of Music and Performing Arts, Graz
sontacchi@iem.at

ABSTRACT

Pitch-scale modifications of polyphonic music are usually performed by manipulating the time-frequency representation of the input signal. Most approaches proposed in the past are thereby based on the Fourier transform although its linear frequency bin spacing is known to be inadequate to some degree for analysing and processing music signals. Recently invertible constant-Q transforms (CQT) featuring high Q-factors have been proposed exhibiting a more suitable geometrical bin spacing.

In this paper a frequency domain pitch-shifting approach based on the CQT is proposed. The CQT is specifically attractive for pitch-shifting because it can be implemented by frequency translation (shifting partials along the frequency axis) as opposed to spectral stretching in the Fourier transform domain. Furthermore, the high time resolution of CQT at high frequencies improves transient preservation. Audio examples are provided to illustrate the results achieved with the proposed method.

1. INTRODUCTION

Pitch shifting is a digital audio effect that changes the pitch of a sound or music without altering its duration. That is, all frequencies are scaled by a constant factor. A common approach to pitch-shifting is based on a two-staged process: first, the time-scale of the input signal is modified (time-stretching) and then the output signal is resampled to retain the signals original time-base but having shifted its frequency content. In general, time-scaling can either be performed in the time-domain [1] or the frequency-domain. Approaches operating in the frequency domain are often based on the phase vocoder where time-scaling is achieved by altering the analysis or synthesis frame hop size [2]. In [3] a constant frame-hop phase vocoder is proposed where time-scaling is achieved by copying or deleting frames. Several improvements to phase vocoder-based time-scaling have been proposed to reduce phasiness [4] [5] and transient smearing [6].

Alternatively, several implementations have been proposed that perform pitch-shifting directly without a time-stretching stage, mainly based on the phase vocoder [7] and on synchronous overlap-add (SOLA)[8]. The former operates in the frequency domain while the latter operates in the time-domain. Usually time-domain approaches are more efficient computationally while frequency-domain approaches based on the phase-vocoder achieve higher quality for polyphonic music signals.

Typically phase-vocoder approaches suffer from artefacts which are perceived as phasiness and transient smearing. Both problems stem from the loss of horizontal phase coherence (inter-frame) and/or loss of vertical phase coherence (intra-frame) [2]. The standard implementation to reduce typical phase vocoder artefacts uses instantaneous frequency estimation and phase unwrapping to establish inter-frame phase coherence and a phase locking scheme to (partly) retain intra-frame phase coherence. The phase locking process comprises a spectral peak-picking stage to define regions within each frame that are dominated by single sinusoidal components (peaks). The phase values within these regions of influence are assumed to be dominated by the peak's phase and thus are 'locked' to this phase value after the inter-frame phase propagation has been adjusted.

The quality of the output signal of the phase vocoder-based pitch-shifter strongly depends on whether the implicit assumptions on the time-frequency representation of a particular input signal are valid: The phase update process is only correct when the input signal can be modelled as a sum of a small number of slowly varying sinusoids. Furthermore it is assumed that each sinusoidal component excites a discrete peak in the spectrum and that the spectrum can be divided into *independent* regions that are dominated by a single sinusoidal component. Especially for dense polyphonic music these assumptions do not always hold since phase vocoder-based pitch shifting implementations usually operate on the Short Time Fourier Transform (STFT) representation. A well known disadvantage of the STFT is the rigid time-frequency resolution trade-off providing a constant absolute frequency resolution throughout the entire range of audible frequencies. In contrast to this we know that due to both musical and auditory aspects a frequency resolution is preferred that increases from high to low frequencies (and vice versa for time resolution). Using the STFT representation, two closely spaced sinusoids in lower frequency regions might excite only one spectral peak. On the other hand, the time-resolution at higher frequency regions might be too coarse to capture quick temporal changes.

For phase vocoder-based time-scaling implementations these issues have been addressed by several authors. In [3] a constant frame-rate phase vocoder has been proposed where multiple windows are used in parallel. That is, the input signal is sub-divided

in three frequency bands and a different STFT window length is used in each band (multiresolution FFT). In [5] the authors propose to use a multiresolution technique in the peak-picking stage where the peak detection function is made frequency dependent to address the fact that closely spaced spectral peaks need to be processed separately in lower frequency regions but can be combined in higher frequency regions. This notion stems from the fact that critical bands in the human auditory system are distributed approximately logarithmically and that most audio signals exhibit a non-uniform distribution of their partials.

Unfortunately, for phase vocoder-based pitch-scaling implementations [7] the application of multiple FFT resolutions in parallel is hardly feasible. In this approach each spectral peak (representing a sinusoidal component) and its neighbourhood (region of influence) is shifted in the STFT representation of the input signal along the frequency dimension according to a given pitch shifting factor. The abrupt time-frequency resolution changes of multiresolution approaches proposed for phase vocoder-based time-scaling implementations would not allow for coefficient shifts along the frequency dimension and can thus hardly be applied to frequency domain pitch-shifting implementations. Nevertheless, frequency domain pitch shifting still exhibits several advantages compared to the two-staged time-scaling/resampling implementation: The computational complexity is independent of the scaling factor and sinusoidal components can be shifted independently. That is, single notes in the input signal can be altered while leaving others untouched.

For dense polyphonic music signals, however, a considerable inconvenience that also leads to higher computational complexity is the fact that a different frequency translation (shift) has to be applied on each spectral peak. This is because the STFT frequency bins are linearly spaced whereas scaling all frequencies by a constant factor α corresponds to a constant shift on log-frequency scale.

Applying the constant-Q transform (CQT) to the problem of frequency domain pitch shifting in place of the STFT provides a solution to all of the aforementioned disadvantages of this approach. Constant-Q transform refers to a technique that transforms a time-domain signal $x(n)$ into the time-frequency domain so that the center frequencies of the frequency bins are geometrically spaced and their Q-factors are all equal. In effect, this means that the frequency resolution is better for low frequencies and the time resolution is better for high frequencies. The CQT is essentially a wavelet transform, but here the term CQT is preferred since it emphasizes the fact that we are considering transforms with relatively high Q-factors, equivalent to 12–96 bins per octave. This renders many of the conventional wavelet transform techniques inadequate; for example methods based on iterated filterbanks would require filtering the input signal hundreds of times. The CQT was proposed in [9] but has been playing only a minor role in the fields of music analysis and music processing since then. The main reasons for this were its complexity when broadband music signals are considered and the fact that it lacked an inverse transform that would allow reconstruction of the original signal from its transform coefficients.

In [10] we proposed solutions to these problems providing a Matlab toolbox for efficient computation of the CQT coefficients and reasonable quality reconstruction (around 55 dB SNR) of the original input signal. Recently another approach to invertible constant-Q transforms featuring high Q-factors has been proposed, yielding even perfect reconstruction [11]. The suggested transform is based on frame theory [12] and utilizes nonstationary Gabor frames [13] to achieve geometrically spaced frequency bins and the property of invertibility. In [14] a discrete-time wavelet transform with tunable Q-factor based on a real-valued dilation factor has been presented which is implemented using a perfect reconstruction over-sampled filter bank with real-valued sampling factors.

With efficient invertible CQT implementations now at hand we present a frequency domain pitch shifting algorithm based on the CQT representation of music signals.

First, we briefly describe how frequency-domain pitch-shifting is performed using the Fourier transform and discuss some of its drawbacks. Secondly, we summarize some aspects of the CQT implementation we proposed in [10] that are crucial for the present CQT-based pitch-shifting algorithm outlined in Section 4. In [15] we provide audio examples to illustrate the achieved quality of the proposed approach.

2. BACKGROUND: STFT-BASED FREQUENCY DOMAIN PITCH-SHIFTING

In order to understand the problems that arise in STFT-based pitch-shifting and to provide background for the proposed method, let us first diagnose the frequency domain pitch-shifting implemented using the STFT.

2.1. Implementation

Pitch-shifting using the STFT representation of an audio signal as proposed in [7] is performed in four steps:

1. Peak detection: The simplest scheme consists of declaring that a bin is a peak if its magnitude is larger than that of its two (or four) nearest neighbours. It is assumed that each detected peak represents a sinusoidal component.
2. Define regions of influence: The region of influence is the sub-band around a spectral peak in which it is assumed that all phase values are dominated by the peak's phase. The boundaries of these sub-bands can be defined halfway between two peaks or at the lowest magnitude bin between two peaks.
3. Coefficient shift: Peaks and their regions of influence are shifted by frequency $\Delta\omega_m$, where m is the peak index. As observed in [7], if the relative amplitudes and phases of the bins around a sinusoidal peak are preserved during the translation, then the time-domain signal corresponding to the shifted peak is simply a sinusoid at a different frequency, modulated by the same analysis window.
4. Phase update: Since the frequencies of underlying sinusoids have been changed during the coefficient shift, phase-coherence from one frame to the next is lost. To avoid artefacts due to inter-frame phase inconsistency, phase values need to be updated.

As discussed in Section 1 the standard techniques to retain phase coherence in phase vocoder implementations are instantaneous frequency estimation and phase unwrapping for horizontal phase coherence and phase-locking for vertical phase coherence. Since the loss of phase coherence due to a synthesis frame hop size that differs from the analysis frame hop size (time-scaling) is analogous to loss of phase coherence due to a frequency shift, the same phase update techniques could be used [2]. However, in

[7] a very simple phase updated scheme is proposed that does not involve instantaneous frequency estimation and can thus be implemented very efficiently. If a peak is shifted by $\Delta\omega$, the difference of the peak's phase between two successive frames must be increased or decreased by an amount consistent with the modified frequency of the underlying sinusoid. For a constant-frequency sinusoidal component, phase coherence can be achieved by simply multiplying each STFT coefficient in the region of influence by the complex

$$Z_u = e^{i\Delta\omega_{m,u}R}$$

where R is the frame hop size and $\Delta\omega_{m,u}$ is the frequency difference due to shifting peak m in frame u. These phase rotations have to be accumulated from one frame to the next, that is $Z_{u+1} = Z_u e^{i\Delta\omega_{m,u+1}R}$. Under the assumption that all phase values in the region of influence are only depending on the peak's phase horizontal and vertical phase coherence can thus be retained exactly for a constant-frequency sinusoid. Due to the linear spacing of STFT frequency bins, the phase rotation that has to be applied to the STFT coefficients in this approach is independent of the exact frequency of the sinusoid.

2.2. Drawbacks

The simple phase update process outlined above exploits the fact that DFT frequency bins are uniformly distributed along the frequency dimension. On the other hand, as discussed in Section 1, the rigid time-frequency resolution of the DFT is known to be disadvantageous for processing broadband audio signals. To justify the assumption that each STFT coefficient within a frame can be assigned to one single peak determining its phase value, very long windows need to be applied at lower frequencies. However, this would result in an unacceptable low time resolution for higher frequencies, as rapid temporal changes usually occur at higher frequencies.

In the implementation outlined above we assumed the desired frequency shift $\Delta\omega$ to be known. Obviously this is hardly the case in pitch-shifting applications as frequencies usually need to be scaled by a factor α rather than shifted by a constant frequency $\Delta\omega$. Hence, for each spectral peak the distinct frequency shift corresponding to the desired scaling factor needs to be determined. To avoid relative detuning between different sources or partials in this process, the instantaneous frequencies of spectral peaks need to be estimated which neutralizes the benefit of the simplified phase update approach. Alternatively, the center frequencies of the peak bins could be employed as frequency estimates which again calls for very long STFT windows for lower frequency partials. Another drawback stemming from the linear frequency bin-spacing is the fact that in most cases the desired frequency shifts correspond to a fractional number of DFT bins. That is, usually STFT coefficients need to be interpolated to avoid detuning due to rounding of $\Delta\omega_m$.

When frequency-domain pitch shifting is applied to the entire input signal rather than single sinusoidal components, these issues cause a considerable increase of computational complexity and decrease the overall quality of the pitch-scaled output signal.

In the remainder of this paper we will outline the implementation of a frequency domain pitch-shifting algorithm based on the constant-Q transform in place of the STFT, providing solutions to the above described problems. The pitch-shifting algorithm we propose is based on the CQT implementation we proposed in [10]. Hence, in the next section we will briefly summarize the basic concepts of this implementation. Note that also other implementations [11] could be used as long as they meet the constraints on the time-frequency sampling scheme and intra-frame phase relations we will outline in Section 4.

3. CONSTANT-Q TRANSFORM

To implement frequency-domain pitch-shifting using the CQT toolbox we described in [10] it is not necessary to be aware of all implementation details. Therefore here we will only give an overview about the basic properties of the CQT, introduce user definable parameters and mention implementation aspects that are crucial to understand how pitch-shifting based on the CQT representation of an audio signal can be performed.

3.1. Signal Model

The CQT transform $X^{\mathsf{CQ}}(k,n)$ of a discrete time-domain signal $x(n)$ is defined by

$$X^{\mathsf{CQ}}(k,n) = \sum_{j=n-\lfloor N_k/2 \rfloor}^{n+\lfloor N_k/2 \rfloor} x(j)a_k^*(j-n+N_k/2) \qquad (1)$$

where $k = 1, 2, \ldots, K$ indexes the frequency bins of the CQT, $\lfloor\cdot\rfloor$ denotes rounding towards negative infinity and $a_k^*(n)$ denotes the complex conjugate of $a_k(n)$. The basis functions $a_k(n)$ are complex-valued waveforms, here also called time-frequency *atoms*, and are defined by

$$a_k(n) = \frac{1}{C} w\left(\frac{n}{N_k}\right) \exp\left[i\left(2\pi n \frac{f_k}{f_\mathrm{s}} + \Phi_k\right)\right] \qquad (2)$$

where f_k is the center frequency of bin k, f_s denotes the sampling rate, and $w(t)$, is a continuous window function (here we use the Hann window), sampled at points determined by $\frac{n}{N_k}$. The window function is zero outside the range $t \in [0, 1]$. Φ_k is a phase offset and $\Phi_k = 0$ for the transform we proposed in [10]. C is a scaling factor and

$$C = \sum_{l=-\lfloor \frac{N_k}{2} \rfloor}^{\lfloor \frac{N_k}{2} \rfloor} w\left(\frac{l+N_k/2}{N_k}\right). \qquad (3)$$

The window lengths $N_k \in \mathbb{R}$ in (1)–(3) are real-valued and inversely proportional to f_k in order to have the same Q-factor for all bins k. Since a bin spacing corresponding to the equal temperament is desired, the center frequencies f_k obey

$$f_k = f_1 2^{\frac{k-1}{B}} \qquad (4)$$

where f_1 is the center frequency of the lowest-frequency bin (due to the desired logarithmic frequency resolution there is no DC-bin), and B determines the number of bins per octave. For B = 12 each CQT bin corresponds to one semitone, however, higher values are usually appropriate (in the audio examples, we used $B = 48$). In practice, B is the most important parameter of choice when using the CQT, because it determines the time-frequency resolution trade-off. The corresponding window lengths $N_k \in \mathbb{R}$ are given by

$$N_k = \frac{f_\mathrm{s}}{f_k(2^{\frac{1}{B}} - 1)}. \qquad (5)$$

It is not computationally reasonable to calculate the coefficients $X^{\mathsf{CQ}}(k,n)$ at all positions n of the input signal. To enable

signal reconstruction from the CQT coefficients, successive atoms can be placed H_k samples apart ("hop size"). In order to analyze all parts of the signal properly and to achieve reasonable signal reconstruction, values $0 < H_k \lessapprox \frac{1}{2}N_k$ are meaningful.

3.2. Efficient Computation

Since the direct evaluation of (1) is quite expensive computationally we reduced the complexity by computing the CQT coefficients in the frequency domain and performed the CQT separately for each octave. To understand the data-structure that is produced by the CQT only the latter is important. Figure 1 shows an overview of the CQT transform as computed octave-by-octave, downsampling the input signal by factor 2 when proceeding to the next octave. To facilitate accessing and manipulating CQT coefficients we used a constant hop size $H_k = H$ for all k within the one-octave CQT. Figure 2 shows the resulting sampling of CQT bins in the time-frequency domain. Note that although all frequency bins within an octave are equally sampled, the window lengths N_k are unique for each frequency bin as defined by (5).

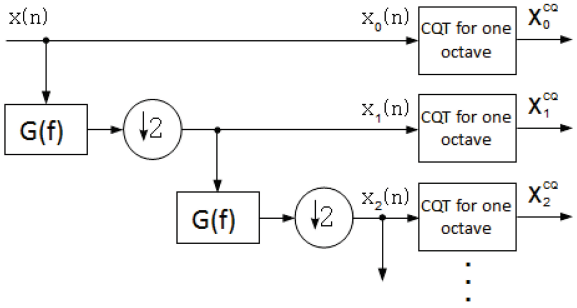

Figure 1: *An overview of computing the CQT one octave at the time. Here $G(f)$ is a lowpass filter and $\downarrow 2$ denotes downsampling by factor two.*

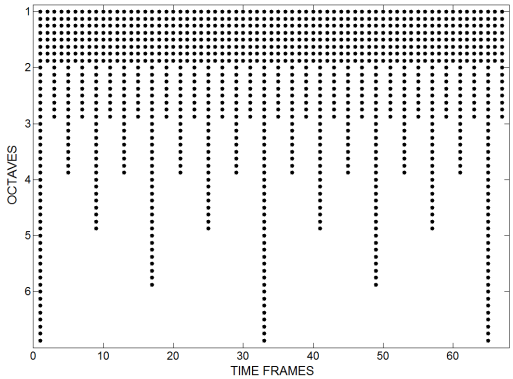

Figure 2: *Points in the time-(log-)frequency plane where $X^{\mathrm{CQ}}(k, n)$ is evaluated.*

3.3. Inverse Transform

Using the toolbox presented in [10] the original signal can be efficiently reconstructed from its CQT coefficients with reasonable quality applying the processing scheme depicted in Figure 1 in reverse order. The signal to noise ratio (SNR) between the original signal and the reconstruction error depends on the CQT resolution, the shape of window function $w(n)$ and the redundancy of the transform.[1] For $B = 48$ and redundancy factors around four or five, the reconstruction quality is around 55 dB SNR.

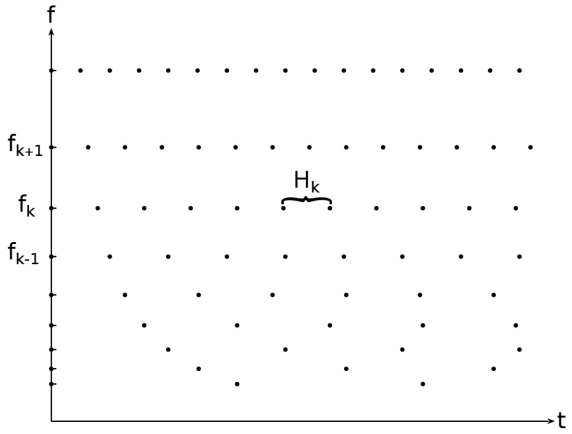

Figure 3: *Minimum redundancy CQT time-frequency sampling scheme. f_k denotes the k^{th} bin center frequency in Hz and H_k is the hop size for channel k in samples.*

4. CQT-BASED FREQUENCY DOMAIN PITCH-SHIFTING

With the efficient and invertible implementation of the constant-Q transform outlined in Section 3 at hand, we will now describe how pitch shifting based on the CQT representation of the input signal can be implemented with ease.

The frequency of a spectral peak in the CQT representation can be scaled by a factor α by translating (shifting) the corresponding CQT coefficient by r CQT bins. For the CQT resolution B (bins per octave) the shift in CQT bins is given by $r = B \log_2(\alpha)$, where r is independent of the frequency of the spectral peak. Furthermore, for the case of chromatic pitch transpositions or transpositions by a fraction of a semitone (e.g. $\frac{1}{8}$-tones for $B = 48$), respectively, r is an integer and no coefficient interpolation is needed. In the following chromatic pitch transpositions will be discussed, however, using simple interpolation arbitrary pitch-scaling factors can be implemented.

4.1. Time-Frequency Sampling Grid

The feasibility of the notion of shifting CQT coefficients along the frequency dimension up- or downwards, however, does not only depend on the placement of sampling points in the frequency domain but also on the sampling scheme in time domain. In Figure 3 an exemplary sampling grid of the time-frequency plane is depicted that produces minimal redundancy of the CQT representation while still being invertible. Such sampling grids are exhibited by CQT implementations where the hop size H_k from one atom to the next (along the time axis) is strictly increasing for increasing

[1] The redundancy factor $R = \frac{2C_{\mathrm{CQT}}}{C_{\mathrm{IN}}}$, where C_{CQT} is the number of CQT coefficients and C_{IN} is the number of input samples.

frequency bin indices k. That is, the overlap factor between successive window functions in time domain is constant while window lengths N_k are decreasing with increasing k. Although producing minimum redundancy, in Figure 3 it can be observed that CQT coefficients cannot be shifted along the frequency dimension without changing its position in time (except for shifts corresponding to one octave). This also means that coefficients need to be skipped or interpolated since the number of coefficients changes from one frequency channel to the next.

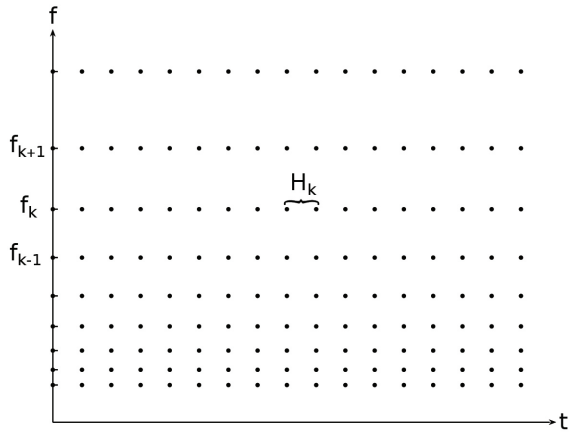

Figure 4: *Rasterized CQT time-frequency sampling scheme. f_k denotes the k^{th} bin center frequency in Hz and H_k is the hop size for channel k in samples.*

One way to overcome this problem is to use a 'rasterized' CQT representation where the hop sizes $H_k = H_K, \forall k \in \{1, 2, ..., K\}$, that is the hop sizes for all center frequencies are set to the smallest hop size in the representation. A time-frequency sampling grid thus obtained is depicted in Figure 4. This sampling scheme yields invertible CQT representations where coefficients can arbitrarily be shifted along the frequency dimension[2]. A major drawback of this sampling scheme, however, is that it produces a highly redundant CQT representation, e.g. for typical transform settings ($B = 48$, $f_1 = 43$ Hz, $f_K = 22050$ Hz, $f_s = 44100$ Hz) the redundancy increases by factor 6.3 compared to the minimum redundancy sampling scheme depicted in Figure 3.

The time-frequency sampling scheme we proposed in [10] can be seen as the middle ground between minimum redundancy and feasibility of coefficient shifts. As discussed in Section 3.2 the atom hop sizes H_k are set according to the highest frequency bin within each octave, that is, for each octave down H_k is multiplied by factor 2. The time-frequency sampling grid thus achieved is depicted in Figure 2 and is reproduced in Figure 5 for the sake of clarity. It can be observed that all CQT coefficients in this representation are temporally aligned, enabling coefficient shifts along the frequency dimension without altering their position in time. For typical transform settings, on the other hand, the redundancy of this CQT representation is only 1.4 times higher than the optimal value. This CQT representation, however, can only be used for pitch shifting factors smaller than 1, that is coefficients shifts towards lower frequencies. For coefficient shifts towards higher frequencies errors due to missing time-frequency sampling points will occur in the frequency area around octave boundaries. To avoid these errors we propose to use an upsampled CQT representation where the number of sampling points in each frequency channel is doubled (except for the highest octave) without changing the window lengths N_k. That is, the redundancy is increased by factor 1.5 (ending up with a redundancy of the representation that is 2.1 times higher than the optimal value) enabling pitch shifting factors up to 2. If even higher factors are desired, higher upsampling factors have to be applied.

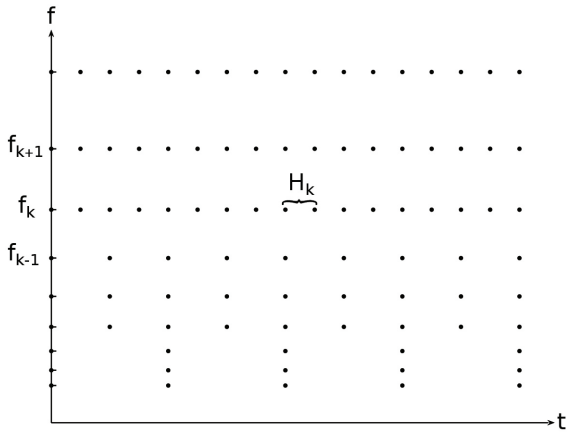

Figure 5: *Synchronized, octave-wise rasterized CQT time-frequency sampling scheme. f_k denotes the k^{th} bin center frequency in Hz and H_k is the hop size for channel k in samples.*

4.2. Phase Coherence

Since sampling points in the time-frequency plane are temporally synchronized, phase coherence from one time instance to the next is lost when CQT coefficients are shifted along the frequency dimension. That is, we need to introduce a phase update stage as is the case with all phase vocoder-based approaches.

4.2.1. Vertical Phase Coherence

The phase locking scheme to retain vertical phase coherence is based on the assumption that the phase relations between a peak bin and its neighbours are invariant under a frequency shift. For a constant-frequency sinusoid, this assumptions holds for the STFT representation in the absence of interfering signal components. The reason for this property of the STFT is, that all DFT atoms are of equal lengths and are all centred at the exact same point within a frame. Hence, the group delays[3] of all DFT bins (band-pass filters) are constant and all equal. This implies that two neighbouring DFT bins that are excited by the same sinusoid (within their main-lobes) will always exhibit a phase difference which is independent of the sinusoid's frequency but is only determined by the common group delay. This fact can be appreciated from figure 6, where $A(\omega)$ is the continuous Fourier transform of $a^{DFT}(n)$, ω_k denotes the center frequency of bin k and ω_x is the frequency of the sinusoidal input signal. In this sketch frame-centred windows are assumed,

[2]This sampling scheme is optional in the CQT implementations [10] and [11].

[3]For the discrete-time discrete-frequency case, the group delay $\tau = -\Delta\Phi(\omega)/\Delta\omega$ where $\Delta\Phi(\omega)$ is the phase difference between two neighbouring bins and $\Delta\omega$ frequency difference between them [16].

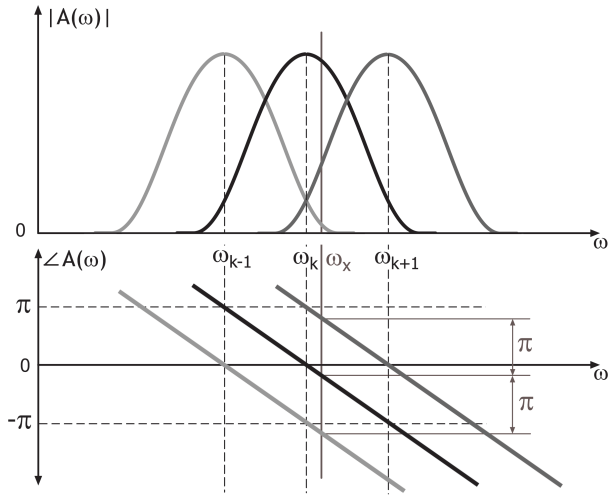

Figure 6: *Continuous magnitude and phase spectra of three adjacent windowed DFT basis functions.*

hence the frequency difference between neighbouring bins is exactly π. Often it is preferred that neighbouring bins excited by the same sinusoid share the same phase value. This is easily achieved by centring $w(n)$ around the first sample of the frame (hence the first and the second half of the input buffer has to be swapped as well).

To establish the same property for the CQT, we have to ensure that all CQT atoms corresponding to the same time instance (*atom stack*) exhibit equal group delays. Hence, the CQT atoms in (2) need to meet two constraints: Firstly, the (symmetric) continuous window function $w(t)$ has to be sampled so that there exists a sample N_c that is located exactly at the window center. Secondly, the phases of the CQT transform basis functions (atoms) $a_k(n)$ have to satisfy

$$\angle a_k(N_c) \stackrel{!}{=} \text{const} \tag{6}$$

for all supported k. Both conditions are met when the CQT transform is implemented according to (1)–(2); however, since N_k is unique for each k in practise they are easily violated. For example if a standard implementation of the (symmetric) Hann window is used the first condition is violated since odd length windows sample the continuous Hann window at the window center and even length windows sample the continuous Hann window around the center. Furthermore, standard window implementations violate the condition in (6) as they do not allow fractional window lengths.

That is, window functions need to be implemented that support fractional window lengths and exact window-center placement for any N_k in order to meet these conditions. Hence we propose to use modified window functions which support arbitrary window lengths and sampling of the window center for all N_k. An implementation of a discrete-time Hann window thus modified is given by

$$w[n] = 0.5\left(1 - \cos\left(\frac{2\pi g_N[n]}{N}\right)\right) \tag{7}$$

where $N \in \Re^+$ is the window length, n is an integer and $0 \leq n \leq 2\lfloor\frac{N}{2}\rfloor$. $g[n]$ is a function that defines where the continuous Hann window is sampled and

$$g_N[n] = \frac{N}{2} - \lfloor\frac{N}{2}\rfloor + n. \tag{8}$$

In Figure 7 four modified Hann windows with different window lengths are depicted. It can be observed that fractional window lengths are supported and that all windows can be exactly stacked at a common center. Note that due to the modified sampling of the continuous Hann window, $w[n]$ is always defined for $2\lfloor\frac{N}{2}\rfloor$ samples.

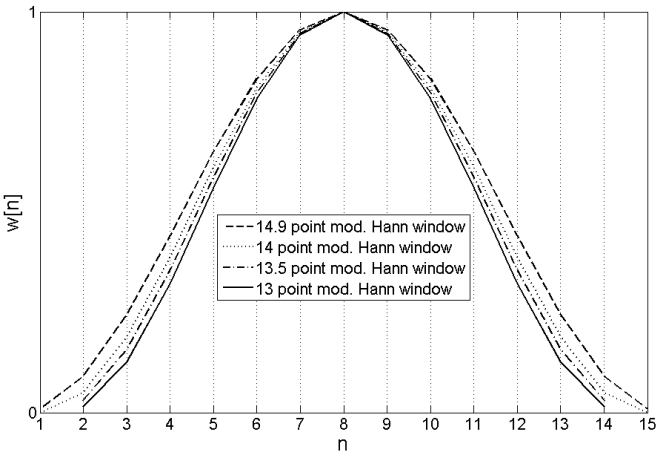

Figure 7: *Modified Hann windows with different (fractional) lengths centred at a common time instance.*

Using the window implementation in (7)–(8), all CQT bins corresponding to the same time instance will exhibit equal phase slopes. For the sake of convenience it might be desirable that an impulse at the atom stack center exhibits real valued (zero-phase slope) CQT coefficients. This can be achieved by setting the phase offset Φ_k in (2) such that $\angle a_k(N_c) = 0$, hence

$$\Phi_k = -\pi N_k \frac{f_k}{f_s} \tag{9}$$

Applying atoms $a_k(n)$ thus implemented, neighbouring CQT bins excited by the same sinusoid (within their main-lobes) will exhibit equal phase values and vertical phase coherence can be retained via phase-locking after translating the CQT coefficients.

4.2.2. Horizontal Phase Coherence

To achieve horizontal phase coherence we need to account for the frequency difference $\Delta\omega = \omega_x^a - \omega_x^b$, where ω_x^b and ω_x^a are the peaks instantaneous frequency before and after the coefficient shift, respectively. The instantaneous frequency ω_x^b is usually obtained from the frame-to-frame (unwrapped) phase difference [2]. In Section 2, however, we outlined the simple phase update scheme proposed in [7] that circumvents the need for instantaneous frequency estimation. Here the phase update is solely based on the frequency difference $\Delta\omega_p = \omega_{k2} - \omega_{k1}$, where ω_{k1} is the center frequency of the peak bin excited by a sinusoid with frequency ω_x^b and ω_{k2} is the center frequency of the peak bin after the coefficients have been translated. This technique is valid for the STFT based approach since $\Delta\omega_p = \Delta\omega$ due to the linearly spaced frequency bins. As CQT bins are geometrically spaced, this is not true for the CQT based approach in general. That is, implementing the simple phase update scheme based on the frequency difference $\Delta\omega_p$ rather than the true frequency difference $\Delta\omega$ (thus sparing instantaneous frequency estimation), errors of the horizontal phase

propagation causing slight frequency and amplitude modulations are introduced. To avoid these errors instantaneous frequency estimation can be performed for spectral peaks as proposed in [2].

Informal listening tests, however, suggest that these errors are hardly (if at all) audible in the output signal when compared to the proper frequency estimation technique. One explanation for this is that according to [17], the human hearing is insensitive to amplitude and frequency modulations below certain thresholds[4]. Since the quality of the output signal is hardly degraded when frequency estimation is omitted[5], we recommend to do so as this improves not only computational efficiency but also robustness.

4.3. Implementation

Having described the general idea of CQT-based pitch-shifting and the minor changes that we applied to the CQT toolbox, we will now outline how music signals can be transposed using the CQT:

1. Transform: Using a 1.5 times oversampled CQT representation, the input signal can be transposed in the range of ± 1 octave. The CQT coefficients are presented in the sparse matrix M^{CQT} as depicted in Figure 2. The audio examples were generated using a CQT resolution B of 48 bins per octave.
2. Translating CQT coefficients: The entire input signal is transposed by rotating (shifting) the rows of M^{CQT} up- or downwards. Using $B = 48$ a matrix shift by one row corresponds to a transposition of 25 cents (quarter of a semitone).
3. Phase update: Using a simple peak-detector for all valid CQT coefficients within the columns of M^{CQT}, each column (frame) is divided into several regions of influence. Horizontal and vertical phase coherence can be retained by multiplying all coefficients within the same region with the complex $Z_u = e^{i\Delta\omega_{k,u}H_k}$ where $\Delta\omega_{k,u}$ is the difference between the center frequencies of the old and the new peak bin in column u of the CQT and H_k is the atom hop size. The applied phase rotations need to be accumulated from one frame to the next.
4. Inverse Transform: Reconstruct the transposed output signal from the processed CQT representation.

5. RESULTS

Up to this point we have not conducted formal listening tests to objectively assess the achieved quality of the proposed pitch-shifting algorithm. However, in [15] we provide several audio examples to demonstrate the performance for pitch-shifting factors in the range of ± 1 octave for different music genres. From these samples it can be appreciated that, despite the very simple implementation (no instantaneous frequency estimation, no peak-following [3], no trajectory heuristics [5], no transient detection [19]), the proposed algorithm produces very little artefacts for a wide range of scaling factors. Due to the logarithmic frequency resolution of the CQT, relative detuning between partials is avoided and closely spaced sinusoidal components at lower frequencies can be distinguished. In [5] a STFT-based phase vocoder for time-scaling has been proposed that includes a frequency dependent peak-picking stage to reduce artefacts in the output signal. Due to the geometrical bin spacing, frequency dependent peak-picking is inherent in the proposed CQT-based pitch-shifting technique.

As discussed above a major benefit of the proposed method is that polyphonic music signals can be transposed in the time-frequency domain by simply shifting the entire CQT up- or downwards (followed by a phase-update stage). An advantage of time-frequency domain pitch shifting approaches in general is that individual signal components (e.g. single notes) can be transposed in polyphonic music signals while leaving others unchanged. The CQT-based approach is specifically interesting for this application since harmonic structures can be easily detected (distribution of partials does not depend on the fundamental frequency) and interference among fundamental frequencies in lower frequency areas is reduced.

5.1. Transients

In general transients pose a problem for all phase vocoder-based approaches for time- and pitch-scaling as the phase update process only considers slowly varying sinusoidal components, hence transients are usually softened (smeared) due to loss of vertical phase coherence at transients. For time-scaling algorithms based on the STFT phase vocoder different solutions to this problem have been suggested in the past. In [19] a transient preservation technique is proposed where phases are reset to their original values at transients. Another approach [3] suggests to set the time-scaling factor to 1 in transient regions (this is compensated by increasing the time-scaling factor in steady-state parts). Both approaches need to rely on a transient detection stage of some kind.

The CQT-based pitch shifting approach mitigates problems with transients by providing a very good time resolution at high frequencies. Therefore transients are preserved simply due to the high time resolution of the magnitude CQT spectrum, without the need to encode the transients in vertically synchronous phase information. However, at low frequencies atom lengths N_k get very wide and the lack of vertical phase coherence gets increasingly audible towards lower frequencies. That is, low frequency transients (e.g. electric/upright bass notes, bass drum hits) call for dedicated processing to reduce low frequency transient smearing.

To avoid the need for explicit transient detection we suggest to include a percussive/harmonic separation technique that only operates in lower frequency regions. As transients do not carry tonal information, low frequency transients can be subtracted from the input signal and added back to the output signal after pitch-shifting. In [20] a simple and efficient percussive/harmonic separation approach has been proposed where transients are regarded as outliers along the time-dimension and harmonic components are regarded as outliers along the frequency dimension in the STFT representation of the input signal. Percussive and harmonic signal components are separated by applying 1-dimensional median filters both along the frequency- and the time-dimension.

To subtract low frequency transients from the input signal we implemented a straightforward CQT adaptation of the technique proposed in [20] that only considers CQT coefficients below the frequency threshold f_{th} since higher frequency transients do not need to be processed. In [15] we provide audio examples to demonstrate that using this efficient approach, low frequency transients can be retained in the pitch-scaled output signal.

[4] This fact has also been exploited in [18] to reduce phasiness for time-scaling applications of STFT-based phase vocoders.

[5] Formal listening tests to proof this result are yet to be conducted.

5.2. Limitations of the method

To obtain a natural sounding pitch-shifted output signal it is desired to retain the formant structure of the original as formants are independent of the fundamental frequencies. A common approach is to model the spectral envelope, use this model to flatten the magnitude spectrum of the input signal and apply the original spectral envelope to the pitch-shifted signal. Several techniques to gain an estimate of the spectral envelope have been proposed, most prominently approaches based on linear prediction [21] or the real cepstrum [22]. Up to this point we have not included any formant preservation technique in the CQT-based pitch-shifter, that is the audio samples we provide in [15] are lacking naturalness (especially when singing voice or speech is considered). However, since formants approximately feature the constant-Q property (i.e. the bandwidth of high frequency formants is wider than for lower frequency formants), we expect the CQT-based pitch-shifting algorithm to exhibit some advantageous qualities for formant preservation approaches in future implementations.

Especially when speech signals are to be transposed, a loss of naturalness does not only occur due to formant shifts but annoying artefacts are introduced due to lack of vertical phase coherence among partials. A possible explanation for these artefacts is the fact, that the shape of the underlying glottal pulses changes when the phase relations between partials are altered. In [6] a shape-invariant phase vocoder for speech transformation is proposed that reduces these artefacts. We have not included techniques to retain inter-partial phase coherence, hence when speech signals are transposed applying large scaling factors, audible artefacts are introduced.

6. CONCLUSIONS

A frequency domain pitch-shifting approach based on the constant-Q transform was proposed that exploits the logarithmic frequency bin spacing of the CQT. The presented technique enables pitch-scaling of monophonic and dense polyphonic music signals applying a simple linear translation of the CQT representation followed by a phase update stage. High quality pitch transpositions with large scaling factors can be achieved without estimating instantaneous frequencies of partials. Audio examples have been provided to demonstrate the achieved quality of the proposed algorithm for different scaling factors up to ± 1 octave.

7. REFERENCES

[1] E. Coyle, D. Dorran, and R. Lawlor, "A comparison of time-domain time-scale modification algorithms," in *Proc. 120th Audio Engineering Society Convention*, 2006.

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

[3] J. Bonada, "Automatic technique in frequency domain for near-lossless time-scale modification of audio," in *Proc. of International Computer Music Conference*, 2000.

[4] D. Dorran and R. Lawlor, "Time-scale modification of music using a synchronized subband/time-domain approach," in *IEEE International Conference on Acoustis, Speech and Signal Processing*, 2004.

[5] T. Karrer, E. Lee, and J. Borchers, "Phavorit: A phase vocoder for real-time interactive time-stretching," in *Proc. International Computer Music Conference (ICMC)*, 2006, pp. 708–715.

[6] A. Röbel, "A shape-invariant phase vocoder for speech transformation," in *Proc. Digital Audio Effects (DAFx-10)*, 2010.

[7] J. Laroche and M. Dolson, "New phase-vocoder techniques for pitch-shifting, harmonizing and other exotic effects," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 1999, pp. 91–94.

[8] J. Laroche, "Autocorrelation method for high-quality time/pitch-scaling," in *Proc. IEEE Workshop Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 1993, pp. 131–134.

[9] 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.

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

[11] 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.

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

[13] 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.

[14] I.W. Selesnick, "Wavelet transform with tunable Q-factor," *IEEE transactions on signal processing*, vol. 59, no. 8, pp. 3560, 2011.

[15] C. Schörkhuber, "Pitch shifting using the CQT: Audio examples," Available at `http://www.iem.at/~schoerkhuber/cqt/`, accessed July 09, 2012.

[16] B. Boashash, *Time frequency signal analysis and processing: a comprehensive reference*, Elsevier, 2003.

[17] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and models, 2nd Edition*, Springer Heidelberg, 1999.

[18] D. Dorran, E. Coyle, and R. Lawlor, "An efficient phasiness reduction technique for moderate audio time-scale modification," in *Proc. Digital Audio Effects (DAFx-04)*, 2004, pp. 83–88.

[19] A. Röbel, "A new approach to transient processing in the phase vocoder," in *Proc. Digital Audio Effects (DAFx-03)*, 2003.

[20] D. Fitzgerald, "Harmonic/percussive separation using median filtering," in *Proc. Digital Audio Effects (DAFx-10)*, 2010.

[21] J.E. Markel and A.H. Gray, *Linear prediction of speech*, Springer-Verlag New York, Inc., 1982.

[22] A. Röbel and X. Rodet, "Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation," in *Proc. Digital Audio Effects (DAFx-05)*, 2005.