The least-squares invertible constant-Q spectrogram and its application to phase vocoding

A. N. Ingle[a] and W. A. Sethares
Department of Electrical and Computer Engineering, University of Wisconsin-Madison, 1415 Engineering Drive, Madison, Wisconsin 53706

(Received 30 December 2011; revised 7 June 2012; accepted 12 June 2012)

This paper discusses the development of a constant-Q spectrogram representation that is invertible in a least-squares sense. A good quality inverse is possible because this modified transform method, unlike the usual sliding window constant-Q spectrogram, does not discard data samples when performing the variable length discrete Fourier transforms on the signal. The development of a phase vocoder application using this modified technique is also discussed. It is shown that a phase vocoder constructed using the least-squares invertible constant-Q spectrogram (LSICQS) is not a trivial extension of the regular FFT-based phase vocoder algorithm and some of the mathematical subtleties related to phase reassignment are addressed. © *2012 Acoustical Society of America*. [http://dx.doi.org/10.1121/1.4731466]

PACS number(s): 43.60.Hj, 43.75.Zz [SAF] Pages: 894–903

I. INTRODUCTION

A. Least-squares invertible constant-Q spectrogram (LSICQS)

Conventional methods for analyzing signals in the frequency domain often use the Fourier transform (FT) or the discrete Fourier transform (DFT), which can require considerable care when dealing with audio signals that vary significantly over time. Information is localized by extracting a section of the time signal via windowing; this leads to a fundamental tradeoff between good time resolution and frequency resolution. There are three ways to bypass this problem: the use of variable windows, the wavelet transform, and the reassigned spectrogram,[1] which subsumes a variety time-frequency analysis techniques and is especially suited to speech signals. This paper uses a variable window DFT that provides direct interpretability of phase values and allows tractable implementation of a phase vocoder.

Variable length windows are fundamental to the constant-Q transform of Brown.[2] Each window is "tuned" to a particular frequency characterized by the Q factor, which can be understood as a ratio of the center frequency to the frequency resolution. A disadvantage of this method is that it is not invertible due to the temporal and frequency decimations. The present work extends the variable length windowing method to achieve least-squares invertibility of the spectrogram. In a sequel, Brown also discusses an efficient algorithm for calculation of the constant-Q transform[3] using the fast Fourier transform (FFT) algorithm. The LSICQS introduced here is formulated as a linear transform operator that produces a vectorized spectrogram directly from the time domain signal.

One way applying a nonlinear mapping to the frequency axis is through the use of the warped FT or the warped DFT,[4] which applies an all-pass filter in the time domain to achieve a (typically trigonometric) nonlinear stretching of the frequency axis. The nonlinearities are tightly constrained and may not always be suited to audio signals. In music signals, most of the sound energy is localized around frequency bands that are geometrically spaced in powers of 2 as this octave structure forms the basis of the musical scale in Western music and closely approximates the octave structure found in other music cultures too. Furthermore, psychoacoustic experiments have revealed that the response of the human ear to sound is, to a first approximation, constant-Q.[5] Hence what appear to be equispaced pitches are really equispaced on a log-frequency axis. Mimicking this perception on a spectrogram is more difficult than merely stretching linear-frequency DFT data so they fit on a log scale.

As a modification to Brown's formulation, Bradford *et al.*[6] suggest aligning the variable length constant-Q windows with the center sample of the signal allowing the variable length windows to analyze the "same part of the signal." This alignment is also used when setting up the LSICQS while bypassing the temporal decimation issue to allow invertibility.

Inspired by a problem in geophysics, Stockwell[7] suggested using Gaussian windows of widths inversely proportional to the frequency when calculating the continuous-time STFT. An analogous strategy that is not limited to any specific window type is used in the LSICQS, allowing applications where geometric spacing between frequencies is important. This plays a crucial role in the phase adjustment strategy in the LSICQS phase vocoder.

Gambardella[8] observes similarities between the constant-Q transform and the Mellin transform. However, his work only focuses on the long-time constant-Q transform and makes no comments on the invertibility of the short-time transform that is of interest when localizing information in both frequency and time.

FitzGerald *et al.*[9] adopt an optimization approach for inverting the constant-Q transform by first mapping the

[a]Author to whom correspondence should be addressed. Electronic mail: ingle@wisc.edu

 0001-4966/2012/132(2)/894/10/$30.00 © 2012 Acoustical Society of America

Author's complimentary copy

constant-Q transform to the DFT domain and then inverting to the time domain. In certain nice cases, the DFT representation is sparse and can be obtained by ℓ_0 or ℓ_1 norm minimization. In the present work, it is shown that with a modified approach for calculation of the spectrogram, the standard least squares technique can be used for inverting directly to time domain.

In a recent paper, Schorkhuber and Klapuri[10] propose an efficient constant-Q computational toolbox and develop a better quality reconstruction technique. They avoid the use of wide windows at low frequencies by processing each octave and then downsampling by a factor of 2. The octave-based transform is formulated as a matrix operation using a spectral kernel and inversion is done by reversing these steps, first using the inverse spectral kernel followed by upsampling. The LSICQS also exploits the idea of using a matrix operator, but with a formulation that enables analysis of all time samples by every window.

B. Phase vocoders

Many audio editing and effect-insertion techniques operate by modification of the sound spectrogram; the edited spectrogram is then inverted back to the time domain. Spectrogram magnitudes are easily interpreted, while phase values are harder to control and alter. The phase vocoder builds phase values back into the edited spectrogram so that the magnitude peaks in adjacent spectral frames connect smoothly. It maintains frequencies by adjusting the phases over time proportional to the particular frequency and maintains continuity between audio segments by smoothening out abrupt variations in phase that may otherwise lead to clicks or discontinuities.

The term "phase vocoder" originated in the late 1930s as a method of encoding and decoding voice,[11] although today it is used for any technique that is capable of operating on the magnitude and phase values in a time-frequency representation and reconstructing a meaningful audio signal from the modified spectrogram. Flanagan and Golden[12] describe a continuous-time version of a phase vocoder that analyzes speech signals using short time phase and magnitude spectra.

Dolson's tutorial[13] presents two mathematically equivalent interpretations of phase vocoding as a filter bank and as a FT. Laroche and Dolson[14,15] study the behavior of various phase assignment techniques based on the observation that phase values must be adjusted to be consistent with the corresponding frequency and the time difference between adjacent windows. Adapting these ideas in the LSICQS phase vocoder is complicated by the fact that the frequencies are spaced geometrically instead of the usual linear scale in a DFT.

Puckette[16] suggests a "phase locking" technique by assuming that the phase values at various DFT bins are locked in a definite way to the phase at the magnitude-peak bin. This can be explained based on the phase profiles of FTs of commonly used window functions. This idea forms the core of the phase assignment strategy presented in this paper, and it will be seen that some mathematical insight is needed to characterize the phase locking behavior in the constant-Q case.

Traditionally phase vocoders have been designed to operate on FFT-based spectrograms with a linear frequency axis. The key idea in the LSICQS phase vocoder is a spectrogram editing technique capable of operating on the constant-Q spectrogram. It is not far-fetched to expect that editing on a log-frequency spectrogram may produce results that are more attuned to the perceptual mechanism than those obtained when using linear frequency spacing. For example, in an audio morphing application, it would be suitable to morph between frequencies that are "nearby" on a log-frequency scale rather than linear spacing.

As it will become clear in Sec. III, the LSICQS phase vocoder is not a trivial extension of the FFT-based phase vocoder because there are some mathematical and algorithmic subtleties involved in its implementation.

A previous attempt by Garas[17] at implementing the constant-Q phase vocoder was controversial. The present work overcomes its shortcomings by implementing the phase vocoder on the LSICQS instead of a regular constant-Q spectrogram. The results discussed in this paper were obtained from a computer implementation of the LSICQS phase vocoder that draws ideas from previous implementations of the regular phase vocoder by Moller-Nielson[18] and Sethares.[19]

II. LEAST-SQUARES INVERTIBLE CONSTANT-Q SPECTROGRAM

A. The constant-Q transform

The constant-Q transform performs calculations directly on a log-frequency scale. For musical signals, the natural choice of frequencies is those in an equitempered scale.

A fixed-length DFT gives constant resolution at all frequencies. For instance, a window that is 1024 samples long and is sampled at $f_s = 44.1$ kHz gives a frequency resolution of about 43.1 Hz that is too large to detect the difference between low notes on a piano, yet this resolution is wasteful at high frequencies.[2]

A more parsimonious approach is to maintain a constant ratio of center frequency to frequency resolution by choosing different window lengths at different frequencies. This ratio is the "Q" of the transform given by

$$Q = \frac{1}{2^{1/\lambda} - 1} \tag{1}$$

where λ is the number of bins per octave. For example, when $\lambda = 48$ bins/octave, $Q \approx 67$. Any frequency f_k associated with a resolution δf_k requires a window length of $N_k = f_s/\delta f_k = Q f_s/f_k$. The kth coefficient in an N_k-length DFT is[2,3]

$$X_k = \frac{1}{N_k} \sum_{n=0}^{N_k-1} w(k, n) x(n) e^{-j2\pi Q n/N_k} \tag{2}$$

where $x(\cdot)$ is the sampled sequence and $w(\cdot,\cdot)$ is a window function.

Author's complimentary copy

Observe that Eq. (2) does not analyze all data samples at high frequencies where the window length is small (time decimation), causing the non-invertibility of the constant-Q transform. A constant-Q spectrogram is obtained by stacking columns of constant-Q transforms from adjacent time segments, analogous to the way the STFT spectrogram is formed by stacking columns of DFTs. To localize frequency information at a particular time, the constant-Q spectrogram extracts a small piece (called a "time slice") from the signal and calculates the constant-Q transform of this slice. For this to work, there must be at least as many samples in the time slice as the longest window that is used when calculating the constant-Q transform. It may also be desirable to use a certain fraction of overlap between adjacent pieces as is common with the STFT spectrogram.

Exact interpretation of the adjacent window overlap factor is complicated in the constant-Q spectrogram because there are two kinds of windowing operations occurring in the evaluation of the spectrogram. The first operation is that of extracting a time slice, whereas the second windowing operation occurs when the constant-Q transform of this time slice of the signal is calculated using variable length windows. Hence, in reality, there is a variable amount of overlap between the actual analysis windows. The longer windows will have a larger overlap, whereas the smaller windows may not overlap at all. This issue is resolved in the LSICQS as described in the next section.

B. The LSICQS in matrix form

The LSICQS is obtained from a time-domain signal by the use of frequency-dependent variable length windows as in the constant-Q transform. However, every window is forced to analyze all data samples, thereby overcoming the temporal decimation issue.

To handle the data structure generated by the LSICQS transformation process, it is convenient from a computational point of view to vectorize the transform.

To arrive at a matrix representation of the transform process, consider how a particular coefficient in the LSICQS is generated. Let $x(n)$ be the time domain signal. For given analysis frequency f_k with an associated constant-Q window length N_k, the first element in the kth row of the LSICQS is generated using the inner product

$$X_0^{f_k} = \sum_{n=0}^{L-1} w_0(n)x(n) \tag{3}$$

where L is the length of the data sequence and $w_0(n)$ is the zero padded windowed complex exponential given by

$$w_0(n) = \begin{cases} v(n)e^{-j2\pi Qn/N_k}, & \text{if } 0 \le n \le N_k - 1 \\ 0, & \text{if } N_k \le n \le L-1. \end{cases}$$

Here $v(n)$ is any appropriate window function.[20]

The subsequent entries in this row of the spectrogram can be obtained by repeating the inner product of Eq. (3) using circularly rotated versions of the $w_0(n)$ vector while maintaining the required overlap. If $w_i(n)$ is the vector obtained after i circular shifts of $w_0(n)$, the ith element of this LSICQS row can be calculated as

$$X_i^{f_k} = \sum_{n=0}^{L-1} w_i(n)x(n). \tag{4}$$

The circularly shifted versions of the $w_0(n)$ vectors corresponding to the analysis frequency f_k can be stored in the rows of a matrix $\mathbf{A}_k$. Hence the kth row of the LSICQS of the time domain data vector $\mathbf{x}$ can be obtained through the linear transformation

$$\mathbf{A}_k\mathbf{x} = \mathbf{b}_k$$

where $\mathbf{b}_k$ is the vectorized form of the kth row of the LSICQS. The fact that many entries in the $\mathbf{A}_k$ matrix are zero can be used to speed up this matrix multiplication (for instance, by storing it as a sparse matrix). Note that the number of rows r_k in $\mathbf{A}_k$ depends on the length of the original signal L, the size of the window N_k, and the overlap fraction p, via the relation

$$r_k = \left\lceil \frac{(L - N_k)}{N_k(1-p)} \right\rceil \tag{5}$$

where $\lceil z \rceil$ is the smallest integer greater than or equal to z. Next, all the $\mathbf{A}_k$ matrices can be stacked to form another matrix, $\mathbf{A} = [\mathbf{A}_0^T|\mathbf{A}_1^T|\cdots|\mathbf{A}_k^T|\cdots]^T$. This matrix, when operated on the time domain signal $\mathbf{x}$, produces a vectorized form of the entire spectrogram. It is important to choose the transform parameters so that the total number of rows in $\mathbf{A}$ exceeds the length of $\mathbf{x}$, otherwise the inverse problem is ill-posed. The complete operation can now be compactly represented as a matrix multiplication, $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{b} = [\mathbf{b}_0^T|\mathbf{b}_1^T|\cdots|\mathbf{b}_k^T|\cdots]^T$ is the vectorized form of the LSICQS. In summary, the transform matrix is constructed by stacking up submatrices, each constructed from a collection windowed complex exponentials tuned to a particular analysis frequency. The structure of one such submatrix is represented in Fig. 1.

C. Consequences of the peculiar structure of the LSICQS and invertibility

The LSICQS uses a sliding window with a preset percentage overlap between adjacent windows to obtain the spectrogram directly from a data sequence. For every frequency of interest, there is an associated window length as dictated by the constant-Q. Because higher frequencies have smaller windows and lower frequencies use longer windows, the sliding windows produce fewer coefficients at lower frequencies than higher frequencies, unlike the usual STFTs, where an equal number of spectrogram points are obtained for each frequency. This allows the use of constant percentage overlap irrespective of window size.

Author's complimentary copy

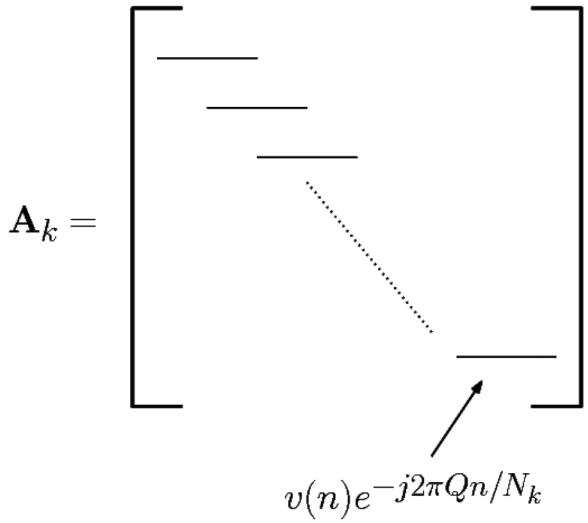

FIG. 1. Structure of a submatrix that is used for analyzing a particular frequency. The full transform matrix is constructed by stacking such submatrices, one for each frequency of interest.

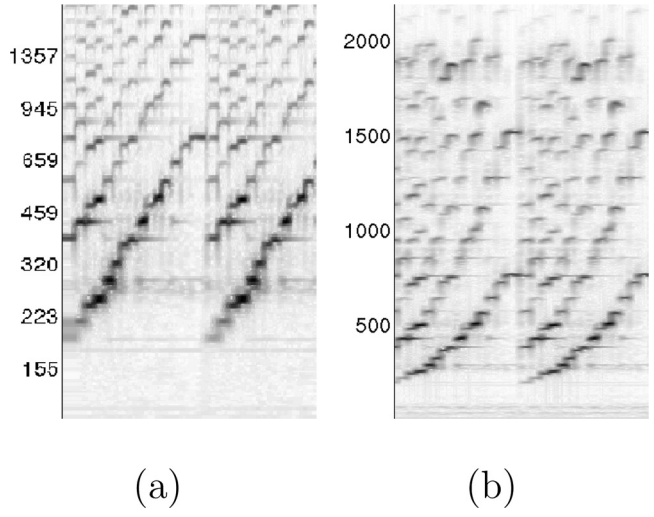

FIG. 3. (a) LSICQS of a violin scale. Successive tones that are equally spaced in a perceptual sense are equally spaced visually. (b) STFT of the same violin scale. Successive tones that are equally spaced in a perceptual sense are compressed visually in the lower frequencies and expanded visually in the higher frequencies.

When viewed in the time-frequency plane, instead of generating a uniformly spaced rectangular grid of numbers, the LSICQS produces non-uniformly spaced points, where time points are linear but non-uniform and frequencies are log-spaced as shown in Fig. 2.

An immediate consequence of this non-rectangular LSICQS structure is that it cannot be directly displayed as an image. The LSICQS data can be interpolated to obtain a uniform grid in the time-frequency plane as shown in an example in Fig. 3(a). Figure 3(b) shows the STFT of the signal for comparison.

The inverse problem for the matrix formulation can be posed as the unconstrained least squares optimization

$$\min_{\mathbf{x}\in\mathbb{R}^L} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2$$

where $\mathbf{A} \in \mathbb{C}^{M\times L}$, $\mathbf{b} \in \mathbb{C}^M$, L is the length of the data vector, and M is the number of rows in $\mathbf{A}$. Although there is no guarantee that $\mathbf{A}$ is full rank, a least-squares minimum norm solution can always be calculated. In some sense, this inverse merely returns a signal that is close to the original signal in ℓ^2 sense and is composed of a weighted sum of frequencies spaced in powers of 2, at user specified log-resolution. In practice, this gives satisfactory results when the audio signal consists largely of pitched musical instruments.

At first sight the inverse problem may appear to be a *constrained* least squares optimization problem because $\mathbf{A}$ and $\mathbf{b}$ are complex valued, whereas the data vector $\mathbf{x}$ must be real valued. However, this constraint can be removed by decomposing into real and imaginary parts.

Let $\mathbf{A} = \mathbf{A}_R + j\mathbf{A}_I$ and $\mathbf{b} = \mathbf{b}_R + j\mathbf{b}_I$ where the subscripts R and I denote the real and imaginary parts, respectively. The goal is to solve for $\mathbf{x} \in \mathbb{R}^L$, which satisfies $\mathbf{A}_R\mathbf{x} = \mathbf{b}_R$ and $\mathbf{A}_I\mathbf{x} = \mathbf{b}_I$ in a least squares sense.

Forming a new augmented kernel matrix $\Lambda = [\mathbf{A}_R^T|\mathbf{A}_I^T]^T$ and an augmented vector $\beta = [\mathbf{b}_R^T|\mathbf{b}_I^T]^T$, which gives the real valued optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^L} \|\Lambda\mathbf{x} - \beta\|^2, \ \Lambda \in \mathbb{R}^{2M\times L} \text{ and } \beta \in \mathbb{R}^{2M}.$$

This can be solved using any standard numerical techniques for unconstrained least-squares problems.

For an example, the top panel in Fig. 4 shows the original audio signal consisting of a violin playing 12 chromatic notes in an octave. The reconstructed audio signal obtained with the inversion technique is shown in the center panel and the absolute difference is shown in the bottom panel. The reconstruction error is on the order of 10^{-15}, which is numerical roundoff error.

This matrix formulation also raises the question of where exactly in time each LSICQS value localizes information. For the sake of convenience, it is assumed that the magnitude and phase information in each LSICQS value corresponds to the center of the window that was used to generate it. The magnitude interpretation seems quite

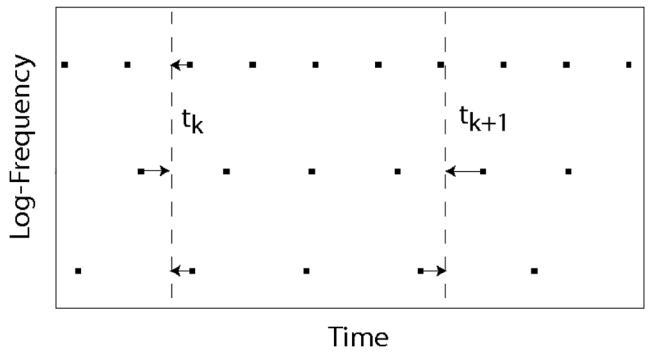

FIG. 2. An exaggerated view of the centers of the windows used for generating an LSICQS. The process of generating pseudo-time-slices by adjusting the phases of the nearest LSICQS bins is also shown (see Sec. III B 1). The arrows indicate the direction of phase adjustment.

Author's complimentary copy

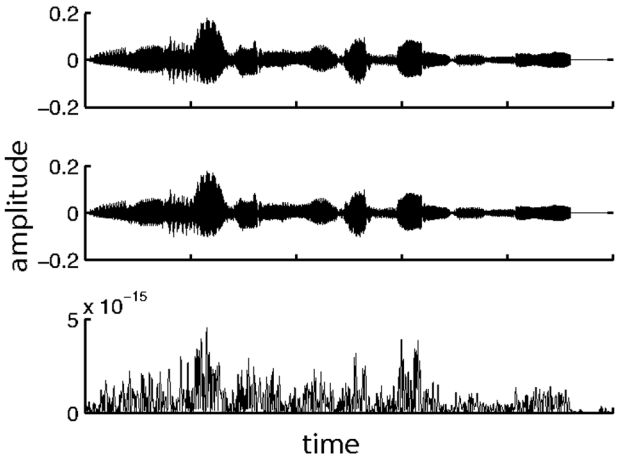

FIG. 4. Example audio signal of a violin playing 12 notes in an octave. Top panel: Original signal. Center panel: Reconstructed signal from the LSICQS. Bottom panel: Time domain reconstruction error showing absolute difference between the original and reconstructed audio signals.

natural; however, it may seem odd that the phase is also referenced to the center of the window (instead of the beginning of the window as it would in a FFT-based spectrogram).

III. IMPLEMENTATION OF A PHASE VOCODER USING THE LSICQS

A. Review of present phase vocoding techniques

The phase vocoder is an analysis-synthesis technique that operates on the spectrogram of the signal and modifies the amplitude and phase values by extracting consecutive time slices from the audio signal. Suppose there is a peak at a particular bin of two consecutive FFT vectors. The phase vocoder estimates the frequency at that peak as

$$f_n = \frac{\theta_2 - \theta_1 + 2\pi n}{2\pi(t_2 - t_1)} \tag{6}$$

where t_1 and t_2 are the reference times at the two adjacent time windows, θ_1 and θ_2 are the phases at the peak-magnitude bins in these adjacent spectral frames. The value of n is chosen so that the estimate f_n is closest to the FFT bin frequency. The estimate obtained in Eq. (6) is better than just picking the frequency at the peak because the energy in FFT bins usually spreads out over multiple neighbors due to spectral leakage.[21]

The modification step assigns new magnitude and phase values to the FFT bins (the exact assignment depends on the audio editing operation). Various schemes of phase assignment have been suggested[14,18,19] to maintain phase coherence during this editing operation. Most of these methods are heuristic and are based on visual analysis of phase profiles in spectrograms of real world signals or by studying the behavior of phase when certain types of tapering-end windows are used. One commonly used phase assignment strategy is the phase-locked vocoder[16] that exploits the property of windowed FFTs that the phase values of bins under the peak are related and "locked" in some way to the phase at the magnitude peak. It is observed that for most commonly used window functions such as Hamming, Hann, and Gaussian, the phases at the bins neighboring the peak are either 0 or π radians offset from the phase at the peak. This zero-π pattern can be observed in the phase pattern of the FTs of these windows. The exact offset depends on the distance from the peak bin. If the phase at the peak bin k is θ_k, a good phase assignment strategy is to use the relation

$$\theta_{k\pm n} = ((\theta_k + \mathrm{mod}(n, 2)\pi))_{(-\pi/2,\pi/2)} \tag{7}$$

to assign the phase at a bin that is n indexes away from the peak.[19] Here $\mathrm{mod}(n, 2)$ is the remainder after dividing n by 2 and $((\cdot))_{(-\pi/2,\pi/2)}$ indicates the operation of wrapping the phase to the interval $(-\pi/2, \pi/2)$.

Finally, the additive-synthesis section of the phase vocoder inverts these FFTs back to the time domain and then overlap-adds the time bursts to generate the modified sound signal.

B. LSICQS phase vocoder

The constant-Q phase vocoder operates on the LSICQS instead of FFTs. The frequency estimation and phase assignment strategies must be modified to account for the log-frequency spacing. Depending on the type of audio editing, the value of Q must be chosen carefully. For instance, it may be desirable to choose a higher Q value if the editing operation involves moving partials from higher octaves into lower octaves so that the frequency detail is still captured.

1. Analysis

The LSICQS phase vocoder algorithm operates on time slices extracted from the LSICQS. However, because the points in the LSICQS are not evenly spaced, a literal time slice does not exist. A "pseudo-time slice" can be constructed by the following correction method. The points that are nearest to the time instant of interest are chosen, and their phases are corrected to reference this instant. This phase correction is proportional to the frequency and the time delta. The magnitudes are set equal to the magnitudes at these nearest points chosen. The analysis section locates the spectral peaks in this pseudo-time slice and estimates the frequency at the peak using the same procedure as a regular phase vocoder in Eq. (6). The construction of a pseudo-time slice is shown in Fig. 2.

In the modification step, new phase values and amplitudes are assigned to the corresponding bins in the new LSICQS. For time scale modification, the analysis chooses an input hop size and an output hop size. The output hop size is taken to be a multiple of the input hop size, where the multiplication factor is equal to the time stretch desired. To achieve time stretching, the magnitudes are unchanged and new phase values are assigned at each peak bin using the relation

$$\theta_k^{(t)} = \theta_k^{(t-\delta t_{out})} + 2\pi f\, \delta t_{out}$$

Author's complimentary copy

where the subscript k denotes the kth peak, the superscript denotes the time at the pseudo-time slice, f denotes the frequency corresponding to this peak as estimated in the analysis step and δt_{out} is the output time hop. Thus a new phase value is assigned so that the frequency appears to unravel the phase through a different time hop to achieve the desired time stretch.

As with the FFT-based phase vocoder, the phase in the region around the spectral peak must be locked to the movement of the phase at the peak. However, unlike the zero-π phase locking strategy used in a regular FFT-based phase vocoder, the LSICQS phase vocoder uses a phase assignment relation that is a consequence of the following theorem.

Theorem 1. Let λ be the bins per octave used for generating a LSICQS with Gaussian windowing and Q be as defined in Eq. (1). Let ω be the frequency at the peak bin and ω_1 be the frequency at a neighboring bin. Then, the first order error term in the difference between phase values of the peak bin and its immediate neighbor is bounded above by $2(\alpha/\pi)^2 1/Q^3$ where α is the parameter used for controlling the standard deviation width of the Gaussian window.

See the Appendix for a proof of this theorem. An analogous result for the Hann window (Theorem 3) can also be found in the Appendix.

To put this result in perspective, consider a typical value of $Q = 34$ (corresponding to $\lambda = 24$ bins/octave) as might be used in practice. Assume $\alpha = 5$. Then the first order error term when using Gaussian windowing is bounded by 1.3×10^{-4}. For the same values, with a Hann window, the bound is 2.5×10^{-5}.

The phase values in the peak region of an LSICQS time slice can be made arbitrarily close to the phase at the peak by choice of the Q factor provided that a suitable window function is used. As a practical consequence of the foregoing theorem, phase assignment under the peak can be done using the simple relation

$$\theta_{k\pm n} = \theta_k. \tag{8}$$

This strategy does result in incorrect phases assigned to bins distant from the peak, but because their magnitudes are small, the phase assigned to those points is unimportant. Observe that this error exists in regular FFT-based phase vocoders also because Eq. (7) holds only for the peak region (and not the complete FFT vector).

The reason for assigning constant phase under the peak can also be understood intuitively. Suppose the signal being analyzed is a single sinusoid of frequency f_k and a constant-Q window of length N_k is tuned to analyze this frequency. The constant-Q window lengths for the adjacent frequency bins f_{k-1} and f_{k+1} are $N_{k-1} = N_k \cdot 2^{1/\lambda}$ and $N_{k+1} = N_k \cdot 2^{-1/\lambda}$, respectively. For typical values of λ that are around 24 or 48 bins per octave, N_{k-1} and N_{k+1} are very close to N_k. The procedure of finding a particular LSICQS coefficient is basically a correlation of a windowed section of the signal with a complex exponential at some frequency. Correlating with the correct window N_k results in a coefficient of largest magnitude and a certain phase value. When the correlation is done with a window for a slightly different frequency, say using N_{k-1} or N_{k+1}, the magnitude of the coefficient drops. However, because the enveloped sinusoid frequency differs only slightly from the actual frequency, the phase offset that gives the best correlation is still close to the phase obtained when the correlation is done with the actual frequency f_k.

This result holds only for "nice" windows such as the Hamming, Hann, Gaussian, and Blackman windows that roll off to negligibly small values near the end. Otherwise the end terms may introduce larger errors in the phase. The analysis of the rectangular window presented in the Appendix quantifies the error incurred.

2. Resynthesis

In the resynthesis step, the edited LSICQS is inverted using the least-squares technique described in Sec. II C.

When the duration of the output signal is not the same as the input signal (as in the time scaling application), it is necessary to use a new matrix $\mathbf{A}'$ with a different overlap factor that achieves the necessary time scaling (and still generates the same LSICQS structure as the original transform matrix). This is akin to stretching the original LSICQS like a rubber membrane to obtain a new time scaled LSICQS that must then be inverted. The new overlap factor can be calculated from the ratio of output to input time hop size, the old overlap factor and Eq. (5). Suppose the signal is to be scaled by a factor s. The goal is to find a new overlap factor p' that gives the same number of rows r_k for this stretched signal of length sL. From Eq. (5), it suffices to have

$$\frac{L - N_k}{(1-p)N_k} = \frac{sL - N_k}{(1-p')N_k}$$

or

$$p' = 1 - \frac{(sL - N_k)(1-p)}{(L - N_k)}. \tag{9}$$

Thus the new overlap factor depends on the frequency index k, indicating that a different overlap factor is needed for each window length N_k. However, assuming that $L \gg N_k$, this dependence on k can be removed to obtain an approximate overlap fraction

$$p' = 1 - s(1-p). \tag{10}$$

This gives rise to "end-effect" errors for a few window lengths at the very end of the analysis duration. This is due to the small differences in the number of rows in the submatrices $\mathbf{A}'_k$ and $\mathbf{A}_k$, causing $\mathbf{A}'$ and $\mathbf{A}$ to have different number of rows. In actual implementation, this can be fixed by changing the size of the edited LSICQS either by appending dummy values or truncating so that the LSICQS row lengths become compatible with the structure of the inversion matrix $\mathbf{A}'$. The final step is to invert the edited LSICQS using the least-squares method with the modified transform matrix.

Author's complimentary copy

Implementation of the analysis-resynthesis sections of the phase vocoder suffers from memory limitations. For very long audio signals, the size of the transform matrix A may become unwieldy and solving the least squares problem difficult. One way of bypassing this is by phase vocoding smaller pieces or frames of the full audio signal and then stitching all the edited frames together. However, additional processing is required to remove discontinuities that may occur at the points where the pieces are stitched. To maintain continuity between adjacent frames and remove clicks, some amount of overlap and windowing needs to be used between adjacent frames. Windows used to specify the frames are henceforth referred to as *overlying windows* and the smaller windows that form a part of the constant-Q transform matrix are called *underlying windows*.

Intuitively, one would expect that the effect of the overlying window will be small if it is sufficiently wider than the longest underlying window. The following theorem mathematically quantifies the effect of this double windowing for the case where both the overlying and underlying windows are Gaussian.

Theorem 2. Let k be the ratio of the width of the overlying to the underlying Gaussian time domain windows, $F(\cdot)$ be the frequency response of the double window and $G(\cdot)$ be the frequency response of the underlying window. Then ignoring any scaling factors, $|F(\cdot)| \to |G(\cdot)|$ pointwise, as $k \to \infty$.

IV. CONCLUSION

This paper presented a novel transform method for obtaining a LSICQS, which can be visualized on a uniform time-frequency grid with an extra interpolation step. The applicability of this spectrogram structure to phase vocoders was discussed, and it was shown that the practical phase assignment strategy follows naturally from a mathematical result that governs the phase values in the LSICQS.

APPENDIX A: DETAILED DERIVATION OF PHASE UNDER THE PEAK

This appendix analyzes phase values of LSICQS coefficients under the peak for various window functions. These mathematical results demonstrate why the constant phase assignment strategy in Eq. (8) works in the case of common tapering-end window functions but not in case of the rectangular window.

A. Proof of Theorem 1 for the Gaussian window

The continuous-time Gaussian window centered at the origin and tuned to the frequency ω_1 with a constant Q is given by

$$w(t) = \begin{cases} e^{-(\alpha^2 \cdot \omega_1^2 t^2/\pi^2 Q)^2}, & \text{if } -\pi Q/\omega_1 \le t \le \pi Q/\omega_1 \\ 0, & \text{otherwise.} \end{cases}$$

Here α is a parameter that is used for controlling the standard deviation, and hence the width of the window in time domain. A smaller value of α gives a larger standard deviation width.

The FT of a Gaussian constant-Q windowed sinusoid, evaluated at a frequency ω_1 is given by

$$\begin{aligned} FT(\omega_1) &= \int_{-\infty}^{\infty} \cos(\omega t + \phi)\, e^{-(\alpha^2 \cdot \omega_1^2 t^2/\pi^2 Q)^2}\, e^{-j\omega_1 t}\, dt \\ &= \frac{\pi^{3/2} Q}{2\alpha\omega_1} e^{-\pi^2 Q^2 (\omega+\omega_1)^2/4\alpha^2\omega_1^2 - j\phi} \left(1 + e^{\pi^2 Q^2 \omega/\alpha^2\omega_1 + j2\phi}\right) \\ &= \left(\frac{e^{-\pi^2 Q^2 (\omega+\omega_1)^2/4\alpha^2\omega_1^2}}{\frac{2\alpha\omega_1}{\pi^{3/2} Q}} \right) e^{-j\phi} \\ &\quad + \left(\frac{e^{-\pi^2 Q^2 (\omega-\omega_1)^2/4\alpha^2\omega_1^2}}{\frac{2\alpha\omega_1}{\pi^{3/2} Q}} \right) e^{j\phi} \\ &=: FT_R(\omega_1) + jFT_I(\omega_1) \end{aligned}$$

where the real and imaginary parts FT_R and FT_I can be easily expressed in terms of the other quantities using Euler's formula.

In terms of the real and imaginary parts of the FT, the phase of the transform at the frequency ω_1 can be written as

$$\angle FT(\omega_1) = \tan^{-1}\left(\frac{FT_I(\omega_1)}{FT_R(\omega_1)} \right). \tag{A1}$$

Consider the behavior of $\angle FT(\omega_1)$ when ω_1 varies around ω by substituting $\omega_1 = \omega + \Delta\omega$. $\Delta\omega$ can be taken as the spacing between the bins under the peak. Defining

$$\epsilon := \frac{\Delta\omega}{\omega},$$

and after some algebraic simplification, the phase function in Eq. (A1) becomes

$$\angle FT(\omega_1) = \tan^{-1}\left(\tan(\phi) \tanh\left(\frac{\pi^2 Q^2}{2\alpha^2(1+\epsilon)} \right) \right). \tag{A2}$$

Treating this as a function of ϵ and writing the first order Taylor series expansion about $\epsilon = 0$ yield

$$\begin{aligned} \angle FT(\epsilon) &= \tan^{-1}\left(\tan(\phi) \tanh\left(\frac{\pi^2 Q^2}{2\alpha^2} \right) \right) \\ &\quad - \frac{\pi^2 Q^2 \sin(\phi)\cos(\phi)}{\alpha^2 \left(\cosh\left(\frac{\pi^2 Q^2}{\alpha^2} \right) + \cos(2\phi) \right)} \epsilon + o(\epsilon^2). \end{aligned}$$

Observe that for large Q, the first term is simply $\tan^{-1}(\tan(\phi))$. Let η denote the absolute value of the first order error term, that is,

$$\eta := \left| \frac{\pi^2 Q^2 \cos\phi \sin\phi}{\alpha^2 \left(\cos 2\phi + \cosh\left(\frac{\pi^2 Q^2}{\alpha^2} \right) \right)} \epsilon \right|.$$

Then, an upper bound can be obtained as follows.

Author's complimentary copy

$$\eta \le \frac{\pi^2 Q^2 |\Delta\omega|}{\omega\alpha^2\left(-1+1+\frac{\pi^4 Q^4}{2\alpha^4}\right)} \tag{A3}$$

$$\le 2\left(\frac{\alpha}{\pi}\right)^2 \frac{1}{Q^3}(2^{1/\lambda}-1) \tag{A4}$$

$$\le 2\left(\frac{\alpha}{\pi}\right)^2 \frac{1}{Q^3} \tag{A5}$$

where Eq. (A3) follows from the inequalities $\cos 2\phi \ge -1$ and $\cosh\gamma \ge 1+\gamma^2/2$; Eq. (A4) follows by considering a bin that is an immediate neighbor of the peak bin and using Eq. (1) and finally Eq. (A5) follows from the fact that $0 < (2^{1/\lambda}-1) < 1$. □

B. Hann window

The phase analysis for the Hann window is quite similar to the Gaussian window case. Moreover, the Hann window result is significant because it can be easily extended to the class of all raised-cosine windows such as Hamming and Blackman window.

The continuous time Hann window centered at the origin and tuned to the frequency ω_1 with a constant Q is given by

$$w(t) = \begin{cases} \dfrac{1+\cos(\omega_1 t/Q)}{2}, & \text{if } -\pi Q/\omega_1 \le t \le \pi Q/\omega_1 \\ 0, & \text{otherwise.} \end{cases}$$

Theorem 3. Consider a LSICQS using a Hann window with all the parameters in Theorem 1. The first order error term in the difference between phase values of the peak bin and its immediate neighbor is upper bounded by $1/Q^3$

Proof. The FT evaluated at a frequency ω_1 using the constant-Q tuned window is given by

$$\begin{aligned} FT(\omega_1) &= \int_{-\pi Q/\omega_1}^{\pi Q/\omega_1} \cos(\omega t+\phi)\frac{1+\cos(\omega_1 t/Q)}{2} e^{-j\omega_1 t} dt \\ &=: \frac{\cos\phi}{2} FT_R(\omega_1) + j\frac{\sin\phi}{2} FT_I(\omega_1). \end{aligned}$$

where

$$FT_R(\omega_1) := \int_{-\pi Q/\omega_1}^{\pi Q/\omega_1} \cos(\omega t)(1+\cos(\omega_1 t/Q))\cos(\omega_1 t) dt$$

and

$$FT_I(\omega_1) := \int_{-\pi Q/\omega_1}^{\pi Q/\omega_1} \sin(\omega t)(1+\cos(\omega_1 t/Q))\sin(\omega_1 t) dt.$$

Define

$$A = \frac{\sin\left(\frac{\pi Q(\omega+\omega_1)}{\omega_1}\right)}{\omega+\omega_1}, \quad B = \frac{\sin\left(\frac{\pi Q(\omega-\omega_1)}{\omega_1}\right)}{\omega-\omega_1},$$
$$C = \frac{\sin\left(\frac{\pi Q\left(\omega+\omega_1+\frac{\omega_1}{Q}\right)}{\omega_1}\right)}{2\left(\omega+\omega_1+\frac{\omega_1}{Q}\right)}, \quad D = \frac{\sin\left(\frac{\pi Q\left(\omega-\omega_1+\frac{\omega_1}{Q}\right)}{\omega_1}\right)}{2\left(\omega-\omega_1+\frac{\omega_1}{Q}\right)},$$
$$E = \frac{\sin\left(\frac{\pi Q\left(\omega_1-\omega+\frac{\omega_1}{Q}\right)}{\omega_1}\right)}{2\left(\omega_1-\omega+\frac{\omega_1}{Q}\right)}, \quad F = \frac{\sin\left(\frac{\pi Q\left(\omega+\omega_1-\frac{\omega_1}{Q}\right)}{\omega_1}\right)}{2\left(\omega+\omega_1-\frac{\omega_1}{Q}\right)}.$$

Then $FT_R(\omega_1) = A+B+C+D+E+F$ and $FT_I(\omega_1) = -A+B-C+D+E-F$ and in terms of these quantities,

$$\begin{aligned} |FT(\omega_1)| &= \frac{1}{2}\sqrt{(FT_R(\omega_1))^2\cos^2\phi + (FT_I(\omega_1))^2\sin^2\phi} \\ \angle FT(\omega_1) &= \tan^{-1}\left(\frac{FT_I(\omega_1)}{FT_R(\omega_1)}\tan\phi\right). \end{aligned} \tag{A6}$$

Mimicking the Gaussian window proof in the previous section, consider the behavior of $\angle FT(\omega_1)$ when ω_1 varies around ω by substituting $\omega_1 = \omega+\Delta\omega$. Defining $\epsilon := \Delta\omega/\omega$ as before, the phase function in Eq. (A6) can be expressed as a function of ϵ. Notice that $0 < \epsilon < 1$ whenever the neighboring frequency bin is close to ω. Taking the Taylor series expansion of $\angle FT(\epsilon)$ about $\epsilon = 0$ yields

$$\angle FT(\epsilon) = \tan^{-1}(\tan(\phi)) - \frac{\sin(\phi)\cos(\phi)}{4Q^2-1}\epsilon + o(\epsilon^2) \tag{A7}$$

where $o(\epsilon^2)$ denotes all the terms containing the second and higher powers of ϵ. It is clear from Eq. (A7) that for small ϵ, the first and higher order terms are negligible, especially when Q is large. Also note that in the LSICQS formulation, the quantities $\Delta\omega$ and Q are coupled via the number of bins per octave (λ) parameter. Choosing a larger value of λ not only makes Q larger but also makes $\Delta\omega$ and ϵ smaller. An explicit bound on the first order error term is

$$\left| -\frac{\sin\phi\cos\phi}{4Q^2-1}\epsilon \right| \le \frac{|\Delta\omega|}{\omega(4Q^2-1)} \tag{A8}$$

$$= \frac{\omega(2^{1/\lambda}-1)}{\omega\left(4\left(\frac{1}{2^{1/\lambda}-1}\right)^2-1\right)}$$

$$= \frac{(2^{1/\lambda}-1)^3}{4-(2^{1/\lambda}-1)^2} \tag{A9}$$

$$\le \frac{(2^{1/\lambda}-1)^3}{3} \tag{A10}$$

$$< \frac{1}{Q^3} \tag{A11}$$

where Eq. (A8) follows from the definition of ϵ and the fact that $|\sin\phi\cos\phi| \le 1$; Eq. (A9) follows by considering the bin that is an immediate neighbor of the peak bin and using Eq. (1) to substitute for Q and finally Eq. (A10) follows from the fact that $0 < (2^{1/\lambda}-1) < 1$. □

Author's complimentary copy

C. Rectangular window

Having shown that the constant phase strategy works for most windows with tapering ends, the behavior with a rectangular window is now analyzed.

$$\begin{aligned} FT(\omega_1) &= \int_{-\pi Q/\omega_1}^{\pi Q/\omega_1} \cos(\omega t+\phi)e^{-j\omega_1 t}dt \\ &= \int_{-\pi Q/\omega_1}^{\pi Q/\omega_1} \cos(\omega t+\phi)\cos(\omega_1 t)dt \\ &\quad - j\int_{-\pi Q/\omega_1}^{\pi Q/\omega_1} \cos(\omega t+\phi)\sin(\omega_1 t)dt \\ &=: FT_R(\omega_1)+jFT_I(\omega_1) \end{aligned}$$

where

$$\begin{aligned} FT_R(\omega_1) &= \int_{-\pi Q/\omega_1}^{\pi Q/\omega_1} \cos(\omega t+\phi)\cos(\omega_1 t)dt, \\ FT_I(\omega_1) &= -\int_{-\pi Q/\omega_1}^{\pi Q/\omega_1} \cos(\omega t+\phi)\sin(\omega_1 t)dt. \end{aligned}$$

In terms of these newly defined quantities,

$$\angle FT(\omega_1) = \tan^{-1}\left(\frac{FT_I(\omega_1)}{FT_R(\omega_1)}\right). \tag{A12}$$

Following the routine, substitute for FT_I and FT_R in Eq. (17), let $\omega_1 = \omega + \Delta\omega$, express $\angle FT(\omega_1)$ as a function of $\Delta\omega/\omega =: \epsilon$ and finally take the Taylor series expansion about $\epsilon = 0$ to get

$$\angle FT(\epsilon) = \tan^{-1}(\tan(\phi)) + \frac{1}{2}\sin(2\phi)\epsilon + o(\epsilon^2). \tag{A13}$$

Observe that the first order error term in Eq. (A13) in case of the rectangular window differs from that in Eq. (A7) for the Hann window by a factor of $(4Q^2-1)$. Typically, Q is at least 34 which indicates a 4000 times magnification in the first order error term when the rectangular window is used. This result explains why the constant phase assignment strategy does not work for such windows.

APPENDIX B: PROOF OF THEOREM 2

Let $k \gg 1$ be the ratio of the widths of the overlying to the underlying Gaussian window. The FT of the underlying window has the form

$$G(\omega) = \frac{1}{\sqrt{\pi}}e^{-\omega^2}$$

and the overlying window has a sharper response of the form

$$H(\omega) = \frac{k}{\sqrt{\pi}}e^{-k^2\omega^2}.$$

where the scaling factors are chosen so that the frequency response functions have unit area.

In the time domain, the overlying window is multiplied pointwise with the underlying window. Moreover, the underlying window need not be aligned to the center of the overlying window. To replicate this, a translation term a is introduced. Using basic properties of the FT, the frequency response of a time shifted version of the underlying window is given by

$$G_{translated}(\omega) = G(\omega)e^{-j\omega a} = e^{-\omega^2 - j\omega a}.$$

Next, using the fact that multiplication in time domain results in convolution in frequency domain the frequency response $F(\omega)$ of the double window can be quantified as follows.

$$\begin{aligned} F(\omega) &= G_{translated}(\omega) * H(\omega) \\ &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi}}e^{-\lambda^2 - j\lambda a}\frac{k}{\sqrt{\pi}}e^{-k^2(\omega-\lambda)^2}\,d\lambda \\ &= \sqrt{\frac{k^2}{\pi(k^2+1)}}e^{-a^2/(4k^2+4) - 4k^2\omega^2/(4k^2+4)}\,e^{-jak^2\omega/(k^2+1)}. \end{aligned}$$

Consider the magnitude of this frequency response function

$$|F(\omega)| = \sqrt{\frac{k^2}{\pi(k^2+1)}}e^{-(a^2+4k^2\omega^2)/(4k^2+4)}. \tag{B1}$$

Clearly, as $k \to \infty$, $|F(\omega)| \to (1/\sqrt{\pi})e^{-\omega^2}$, which is the frequency response of the underlying window. □

Using the Gaussian window makes the algebra in the preceding proof quite tractable. Other window combinations will have qualitatively similar results and the exact properties can be readily analyzed numerically.

[1] S. A. Fulop and K. Fitz, "Algorithms for computing the time-corrected instantaneous frequency (reassigned) spectrogram, with applications," J. Acoust. Soc. Am. **119**, 360–371 (2006).
[2] J. C. Brown, "Calculation of a constant-Q spectral transform," J. Acoust. Soc. Am. **89**, 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. Am. **92**, 2698–2701 (1992).
[4] A. Makur and S. K. Mitra, "Warped discrete-Fourier transform: Theory and applications," IEEE Trans. Circuits Syst., I: Fundam. Theory Appl. **48**, 1086–1093 (2001).
[5] J. G. Roederer, *The Physics and Psychophysics of Music: An Introduction*, 3rd ed. (Springer-Verlag, New York, 2001), pp. 24–28.
[6] R. Bradford, J. Ffitch, and R. Dobson, "Sliding with a constant Q," in *Proceedings of the 11th International Conference on Digital Audio Effects (DAFx-08)*, Espoo, Finland (2008), pp. 363–369.
[7] R. G. Stockwell, L. Mansinha, and R. P. Lowe, "Localization of the complex spectrum: The S-transform," IEEE Trans. Signal Process. **44**, 998–1001 (1996).
[8] G. Gambardella, "The Mellin transforms and constant-Q spectral analysis," J. Acoust. Soc. Am. **66**, 913–915 (1979).
[9] D. FitzGerald, M. Cranitch, and M. Cychowski, "Towards an inverse constant Q transform," in *Audio Engineering Society Convention*, Paris, France (2006), p. 120.
[10] C. Schoerkhuber and A. Klapuri, "Constant-Q transform toolbox for music processing," in *7th Sound and Music Computing Conference*, Barcelona, Spain (2010).
[11] H. Dudley, "The vocoder," Technical Report No. 18, Bell Labs (1939), pp. 122–126.
[12] J. Flanagan and R. M. Golden, "Phase vocoder," Technical Report 45, Bell Systems (1966), pp. 1493–1509.

Author's complimentary copy

[13]M. Dolson, "The phase vocoder: A tutorial," Comput. Music J. **10**, 14–27 (1986).
[14]J. Laroche and M. Dolson, "Phase-vocoder: About this phasiness business," in *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY (1997), 4 pp.
[15]J. Laroche and M. Dolson, "Improved phase vocoder time-scale modification of audio," IEEE Trans. Speech Audio Process. **7**, 323–332 (1999).
[16]M. S. Puckette, "Phase-locked vocoder," in *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY (1995), pp. 222–225.
[17]J. Garas and P. C. W. Sommen, "Time/pitch scaling using the constant-Q phase vocoder," in *Proceedings of ProRISC/IEEE Workshop on Circuits, Systems and Signal Processing*, Utrecht, The Netherlands (1998), pp. 173–176.
[18]P. Moller-Nielsen, "Sound manipulation in the frequency domain," Technical Report, Aarhus University, Aarhus N, Denmark (2002), <http://www.daimi.au.dk/~pmn/sound/> (Last viewed Feb. 18, 2011).
[19]W. A. Sethares, *Rhythm and Transforms*, 1st ed. (Springer-Verlag, London, 2007), pp. 117–121.
[20]J. G. Proakis and D. K. Manolakis, *Digital Signal Processing*, 4th ed. (Prentice Hall, Upper Saddle River, NJ, 2006), pp. 666–667.
[21]M. S. Puckette and J. C. Brown, "Accuracy of frequency estimates using the phase vocoder," IEEE Trans. Speech Audio Process. **6**, 166–176 (1998).

Author's complimentary copy