AN EFFICIENT ALGORITHM FOR REAL-TIME SPECTROGRAM INVERSION

Gerald T. Beauregard
muvee Technologies
g.beauregard@ieee.org

Xinglei Zhu
Institute for Infocomm Research, Singapore
xzhu@i2r.a-star.edu.sg

Lonce Wyse
Institute for Inforcomm Research, Singapore
lonce@i2r.a-star.edu.sg

ABSTRACT

We present a computationally efficient real-time algorithm for constructing audio signals from spectrograms. Spectrograms consist of a time sequence of short time Fourier transform magnitude (STFTM) spectra. During the audio signal construction process, phases are derived for the individual frequency components so that the spectrogram of the constructed signal is as close as possible to the target spectrogram given real-time constraints. The algorithm is a variation of the classic Griffin and Lim [1] technique modified to be computable in real-time. We discuss the application of the algorithm to time-scale modification of audio signals such as speech and music, and performance is compared with other methods. The new algorithm generates comparable or better results with significantly less computation. The phase consistency between adjacent frames produces excellent subjective sound quality with minimal fame transition artifacts.

1. INTRODUCTION

Magnitude and power spectra, and their time sequences in the form of spectrograms, are widely used to represent the time-frequency structure of audio signals such as speech and music. By combining the real and imaginary part of each spectral frequency component into a single number, they provide a valuable visualization tool with a strong correspondence to how signals are heard in terms of frequency content. However, they do so at the expense of information which must be provided in order to convert the representation back into an audio signal.

Frequency magnitude representations are also used computationally in a number of applications (such as noise reduction, signal enhancement, signal source separation etc.), where the frequency domain representation of a signal is modified before being transformed back into a time-domain signal. In general however, a modified (or arbitrary) magnitude spectrum is not a valid representation of an audio signal in the sense that there may be only a complex, but no real signal whose STFTM exactly matches the modified one. In such cases, we would like to find a signal with an STFTM as close as possible to the modified or target STFTM. Griffin and Lim [1][2] developed an iterative least-squares error method for estimating a real audio signal from a modified STFTM (that we abbreviate 'G&L'). Their algorithm monotonically reduces the difference between the target magnitude spectrum (MS) and the MS of the reconstructed signal. The error measure reaches a plateau for most sounds after between 20 and 100 iterations. Though computation is becoming less of an issue as computers get faster, G&L is inherently not real-time since each iteration must loop over all time frames in the signal before the next iteration is computed. As it was originally formulated, the method is unusable for applications with real-time requirements such as noise reduction in real speech transmission, or any application requiring interactive manipulation of the analysis and synthesis parameters. Slaney [11] also developed techniques to reconstruct time-domain audio signals from cochleagrams and correlograms exploiting the G&L technique.

Time-scale modification (TSM) is a process for modifying the rate of signals such as speech or music while keeping other characteristics such as pitch or formants unchanged. TSM is useful in a variety of applications such as music playback. In media production for example, TSM is frequently used to synchronize the audio signal with the video signal. TSM is typically implemented in the time-domain for computational efficiency, but we will demonstrate some advantages to using a frequency domain method.

The rest of the paper is organized as follows. In Section 2 we review the G&L algorithm and show the details of our Real-Time Iterative Spectrogram Inversion algorithm (RTISI) to reconstruct the time-domain signal from a given spectrogram. In Section 3 we briefly introduce the synchronized overlap and add (SOLA) method for time-scale modification and apply the RTISI algorithm to that task as well. In Section 4 we evaluate our method and compare the results with related methods. In Section 5 we make conclusions.

2. RECONSTRUCTION OF TIME-DOMAIN SIGNALS FROM THE MSTFTM

A discrete signal *x(n)* can be represented as a sequence of STFT's as follows:

$$X(mS, \varpi) = \sum_{n=-\infty}^{\infty} x(n) w(n - mS) e^{-j\varpi n} \quad (1)$$

where *w* is the analysis window, *S* is the analysis step size and *m* is the index of the frames of STFT's. The STFT can be considered to be generated by sliding a window *w* across the time domain signal with step size *S*. From $X(mS,\omega)$ we can exactly reconstruct the time-domain signal *x(n)*. However in many applications we need to recover the time-domain from the magnitude spectrum $|X(mS,\omega)|$, or a modified version $|X'(mS,\omega)|$. In Section 2.1 we take a brief look at the G&L method and in Section 2.2 we present the details of the RTISI algorithm.

2.1. The Griffin and Lim (G&L) Algorithm

Starting with an initial estimate $x^0(n)$ of the original time-domain signal $x(n)$, the G&L algorithm iteratively renews the estimate $x^i(n)$ at the i^{th} iteration so that the STFTM of the new estimate is monotonically closer to the STFTM of the original signal $x(n)$ in terms of the distance measure function $D_M[x(n), x^i(n)]$. The distance measure is defined as

$$D_M[x(n), x^i(n)] = \sum_{m=-\infty}^{\infty} \frac{1}{2\pi} \int_{\varpi=-\pi}^{\pi} [|X(mS,\varpi)| - |X^i(mS,\varpi)|]^2 \, d\varpi \quad (2)$$

where $|X(mS,\omega)|$ is the STFTM of original signal $x(n)$ and $|X^i(mS,\omega)|$ is the STFTM of the i^{th} estimate $x^i(n)$.

G&L uses the following function to update the estimate in each iteration,

$$x^{i+1}(n) = \frac{\sum_{m=-\infty}^{\infty} w(n - mS) \frac{1}{2\pi} \int_{\varpi=-\pi}^{\pi} \hat{X}^i(mS,\varpi) e^{-j\varpi n} d\varpi}{\sum_{m=-\infty}^{\infty} w^2(n - mS)} \quad (3)$$

where $|\hat{X}^i(mS,\omega)|$ is the STFT of $x^i(n)$ with the magnitude constraint:

$$\hat{X}^i(mS,\varpi) = X^i(mS,\varpi) \frac{|X(mS,\varpi)|}{|X^i(mS,\varpi)|} \quad (4)$$

By the magnitude constraint, $\hat{X}^i(mS,\omega)$ has the same phase as $X^i(mS,\omega)$ and the same magnitude as $X(mS,\omega)$. A Hamming window with $L=4S$ is used, with scaling such that the sum of the squares of the overlapping windows is always 1. This simplifies the update function in Equation (3), as the denominator will be 1 for all n.

Although Griffin and Lim [1] showed that the distance measure function monotonically decreases with the increasing number of iterations, it was not proven that the algorithm converges to a solution with the globally minimum possible error. Convergence to a local minimum is possible depending on the initial estimate $x^0(n)$. The initial estimate also determines the number of iterations required. In practice, however, within 100 iterations this algorithm generally gives a high quality reconstruction [4].

2.2. The Real-Time Iterative Spectrogram Inversion (RTISI) Algorithm

In order to enable the use of magnitude spectra in interactive and real-time applications, the G&L algorithm needs to be modified so that a given frame is dependent only upon the current and previous frames of the target spectrogram. G&L could also be sped up significantly by finding a better initial estimate of phases for each frame. We accomplish both of these goals by employing a G&L iteration strategy on the current frame alone, using information from the audio frames already reconstructed that overlap with the current frame to construct an initial current frame phase estimate.

Suppose we already have reconstructed the first $m-1$ frames of the synthesis signal, which we denote as $y_{m-1}(n)$. Let us consider the problem of generating frame m. The signal frames already generated at this point are illustrated in Figure 1.

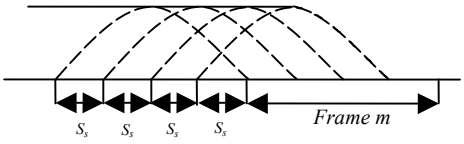

Figure 1. *An illustration of the partially reconstructed frames of signal y(n). Before frame m is estimated, there exists an overlap-added result of the frames m-1, m-2, m-3 in the range of the frame m window. The solid line shows the magnitude contour of the previously synthesized signal and the dashed lines indicates the individual frames.*

As shown in Figure 1, for $m>1$ before we estimate the frame m, the overlap interval is partially filled by the former frames. We use a fixed 75% synthesis window overlap (i.e. $S = L/4$) in our system so that the m^{th} partial frame comes from the overlap-added results of the estimation of the frames $m-1$, $m-2$, $m-3$ of $y(n)$, while the 4th quarter of frame m is all zero. The partial frame will be used to estimate the initial phase in our system as discussed below. To distinguish the partially filled frame from the fully constructed frame m, we notate the former partially filled frame as $y_{m-1}(n)w(n-mS)$, where $w(n)$ is the window function. Now we estimate frame m and overlap-add it with the partial frame $y_{m-1}(n)w(n-mS)$ to generate $y_m(n)$.

To generate an initial estimate for the phases of frame m, we compute the phase of the partially reconstructed signal using an analysis window positioned at the partially constructed frame m. This ensures that even the initial phase estimate for frame m will provide good phase continuity with the partially-reconstructed signal. The Fourier transform of this partial frame is calculated with the same normalized Hamming window as in Section 2.1. We then apply the magnitude constraint of Equation (4) to this Fourier transform keeping the phase unchanged. Next we calculate the inverse Fourier transform of this new frequency domain signal to get a new estimate of frame m. If the maximum iteration number is not reached, we add frame m to the partial frame $y_{m-1}(n)w(n-mS)$, apply the window, and calculate the Fourier transform of the windowed summation to get a new estimate of the phase. We thus use the update Equation (3) from the G&L algorithm in our iterative process but instead of updating the estimate of the whole signal $x(n)$, in each step we update the estimate of the current frame only. The iterative process is illustrated in Figure 2.

There is a special case at the beginning of the signal, where we do not have a partial frame to be added to our estimate. Any initial phase can be used as the initial phase estimation for the first frame. In our experiments we simply use a zero initial phase estimate with the target magnitude spectrum and follow the above iterative process to generate the first frame of $y(n)$.

When the iterative process ends, frame m is combined with the partial frame $y_{m-1}(n)w(n-mS)$ and the process continues with advancing frames until the spectrogram frames are exhausted. We will refer to this method as the Real-Time Iterative Spectrogram Inversion (RTISI) algorithm.

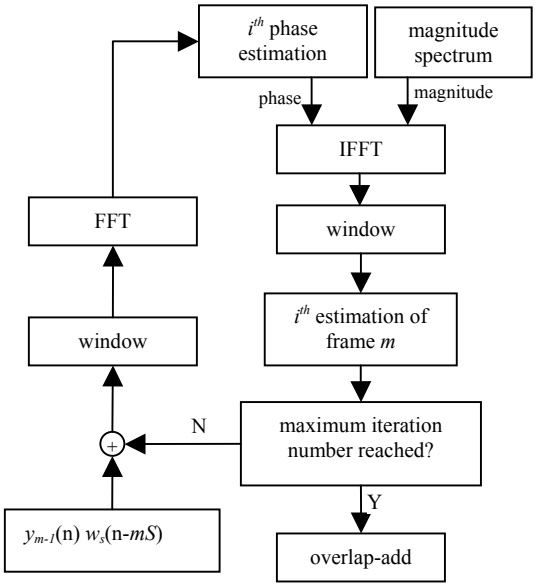

Figure 2. *Frame-by-frame iterative phase estimation process*

Where G&L uses partially reconstructed audio information from frames *m-3* through *m+3* to reconstruct frame *m,* the RTISI method uses information from previous frames only. Also, the partial information used to construct frame *m* with G&L changes with each iteration, whereas in the RTISI method, each frame is estimated strictly in time order.

The frame-by-frame method admits an obvious source of error in that the computation of frame *m* is based only on part of the signal that the target frame *m* is based upon. For this reason, RTISI cannot generally be expected to match the spectral error measure achieved by G&L. The overlap-add procedure adds audio to frame *m* from "future" frames *m+1, m+2* and *m+3* after frame *m* has already been estimated. This will change both the magnitudes and the phases of the given frame when it is analyzed following the complete construction of the signal. This can cause the addition of spectral energy where the magnitude has already been closely matched. However, the future frames overlapping with frame *m* can also compensate for the error resulting from any inability to approach the target magnitude spectrum due to frequency content not present in the partial signal used in the estimation of frame *m.* Thus, the error in the resulting spectrogram is in practice not great, and in any case, inter-frame phase consistency is maintained during future frame overlap, which prevents a common source of perceptual degradation. Furthermore, in the trade-off we gain in speed of convergence given the greater information for the initial phase estimate available using RTISI.

3. TIME-SCALE MODIFICATION OF AUDIO SIGNALS

Time-scale modification (TSM) of signals has long been a subject of interest in the audio and speech processing domain. A key challenge in TSM is to change the audio rate, while preserving other characteristics such as pitch and timbre. There have been several approaches reported to modify the time-scale of an audio signal. Such approaches include the G&L method [1], the synchronized overlap and add algorithm (SOLA) [4] and its various modifications such as WSOLA[5]/SAOLA[6] /PSOLA[7], the phase vocoder algorithm [8] and some methods for building specific models of speech processes such as the vocal tract model[9] and a probabilistic inference model[3].

To achieve time-scale modification, for polyphonic signals the phase vocoder method [8] is a common choice. For monophonic signals, a time-domain process of overlap and add (OLA) is often used as follows. The original signal is first windowed at length L with an analysis step size S_a. Then for each windowed frame a reconstruction signal of the same length is generated and all the regenerated signal segments are overlap-added with appropriate weights for the synthesis step size S_s. However a simple time-domain OLA method does not generally work well because the signal segments being overlap-added may not be consistent when the audio modification rate (S_s/S_a) is other than 1. Different OLA variants modify the basic process to improve quality. For example in the SOLA method, the reconstructed frame varies within a small range to maximize a correlation function to improve the consistency of the scaled result.

The reconstruction of the windowed signal can be implemented in either the time-domain with a method such as SOLA, or in frequency domain with a method such as the phase vocoder or the RTISI algorithm do. Because traditional magnitude-spectra-only reconstruction methods in [1][2] require a large number of iterations of the analysis-synthesis cycle to achieve good performance, the time-domain methods are considered to be economical in computation and have been applied in many commercial implementations. The time-domain TSM methods work well when the modification factor is close to 1 and when the signal source is monophonic, but the performance is often degraded when they are applied to polyphonic sounds or when the modification factor is large [10].

The RTISI method is applicable to both monophonic samples and polyphonic samples. The relatively small amount of calculation required by RTISI and the consistency between the adjacent regenerated frames make it applicable to real-time applications. In Section 3.1 we describe the use of synchronized overlap and add (SOLA) for TSM. In Section 3.2, we discuss the implementation of TSM using the RTISI approach, and then compare the methods in the evaluation section.

3.1. Synchronized Overlap and Add Method

SOLA is a modification of the simple OLA method and it provides more consistency between the reconstructed adjacent frames. Consider a pulse train across two adjacent windowed frames in the source signal. If the TSM rate is other than 1, the pulses in neighbouring frames do not line up for the overlap and add. This creates artifacts such as additional clicks, false frequencies, and reverberation effects. The additional synchronization step in SOLA provides a more consistent solution across frames.

SOLA aligns the adjacent frames in a way that they are of highest similarity where the windows overlap during reconstruction in order to maintain original characteristics of the audio signal such as pitch. It is achieved by sliding the new

analysis frame along the signal reconstructed so far, which is noted as *y(n)*, by an alignment offset k_p in a range of $[k_{min}, k_{max}]$ that maximizes the following normalized cross-correlation function:

$$R_m[k] = \frac{\sum_{q=0}^{L-1} y(mS_s + k + q)x(mS_a + q)}{\sqrt{\sum_{q=0}^{L-1} y(mS_s + k + q)\sum_{q=0}^{L-1} x^2(mS_a + q)}} \quad (5)$$

where L is the length of analysis window, S_s is the synthesis step size, S_a is the analysis step size, m is the index of current frame, and *y(n)* is the reconstructed signal as so far. The SOLA implementation can perform the cross-correlation function of the two signals g1(t), g2(t) efficiently by first taking the Fourier transform G1(w),G2(w) of g1(t) and g2(t) separately and then calculating the inverse Fourier transform of G1(w)*$\bar{G}$2(w), where $\bar{G}$2(w) denotes the complex conjugate of G2(w).

After the best synchronization position k_p is found, the windowed frame m is overlap-added to the synthesis signal with offset k_p and the process is repeated for the next frames until the whole signal is exhausted.

3.2. The RTISI Method for Time-Scale Modification

The way we apply RTISI to time-scale modification follows the traditional frequency domain method: for a modification rate α, we use an analysis step size S_a to obtain the STFTM and use a synthesize step size S_s such that $S_a = S_s/\alpha$. The frame lengths in the analysis and synthesis process are both L. Here we use a fixed synthesis step size $S_s = L/4$, which keeps computational requirements consistent for various modification rates. Given α, we use an analysis step size of S_s/α to achieve the modification rate. The process is shown in Figure 3.

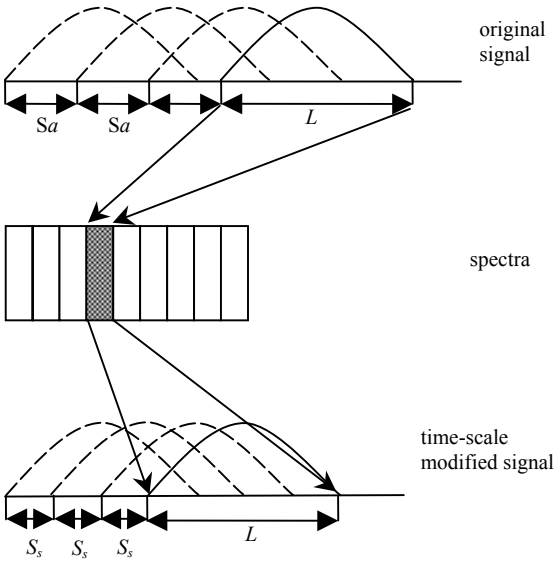

Figure 3. *Time-scale modification in RTISI.*

RTISI can efficiently achieve excellent inter-frame consistency thereby removing the primary obstacle to applying a frequency-domain method to TSM.

4. EVALUATION

The evaluation section is divided into two parts: the evaluation of the phase reconstruction performance and the evaluation of the TSM application.

4.1. Phase Reconstruction Performance

We evaluate the phase reconstruction result using an SNR function similar to the one in [3] comparing the spectrogram of the reconstructed signal to that of the target:

$$SNR = 10\log\frac{\frac{1}{E}\sum_w\sum_f |s_w(f)|^2}{\sum_w\sum_f (\frac{1}{\sqrt{\hat{E}}}\left|\hat{s}_w(f)\right| - \frac{1}{\sqrt{E}}|s_w(f)|)^2} \quad (6)$$

where $|S_w(f)|$ is the STFTM of the original signal, $|\hat{S}_w(f)|$ is the STFTM of the reconstructed signal, $\hat{E}$ is the total energy in the reconstructed signal, summations over w and f are over all windows and frequencies respectively.

We applied the frame-by-frame time-domain signal reconstruction algorithm on a set of 24 test audio samples including chirp samples, frequency modulation samples, pulse train samples, male speech samples, female speech samples and music samples. The average SNR computed by Equation (6) is shown in Table 1 for different numbers of iterations.

Iteration number	1	2	3	4	5	10
Average SNR(dB)	9.25	15.55	16.42	16.62	17.71	18.41

Table 1. *Average SNR measures for RTISI*

Table 2 shows a comparison of RTISI with G&L on the same test set. We run G&L with 5, 10 and 50 iterations, and we run RTISI with 5 and 10 iterations. Figure 4 illustrates the error in the frequency magnitude domain for a typical data (taken from a vowel sound in male speech).

	SNR (dB)
G&L (5 iterations)	10.37
G&L (10 iterations)	12.16
G&L (50 iterations)	15.50
RTISI (5 iterations)	17.71
RTISI (10 iterations)	18.41

Table 2. *Average SNR results for G&L and RTISI on a test set of 24 signals for different levels of iteration.*

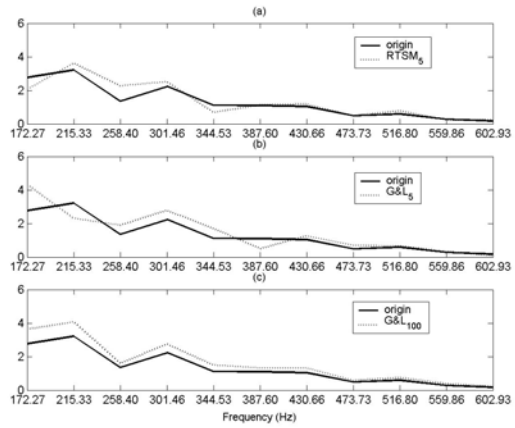

Figure 4. *Magnitude spectra at a particular frame of a male speech vowel. Shown are comparisons between the original magnitude spectra and (a) RTISI at 5 iterations, (b) G&L at 5 iterations, and (c)G&L at 100 iterations.*

Achan[3] recently introduced a method of probabilistic inference for computing an estimate of reconstructed signals given a spectrogram. To compare RTISI with their method, we downloaded the female speech example published by the authors on the website http://www.psi.toronto.edu/~kannan/spectrogram/. Both the original signal and the reconstruction result of the method described in that paper were provided. A comparison of the two methods is shown in Table 3. The SNR figures show some improvement using RTISI, however, there are click artifacts apparent in the result of the Achan method, the perceptual salience of which is not adequately reflected in the SNR numbers.

	SNR (dB)
Achan's method	7.05
Achan with AR model	7.04
RTISI (5 iterations)	10.68
RTISI (10 iterations)	11.86

Table 3. *SNR from different methods for a specific sample*

Figure 5 shows the distance between the original signal and the reconstructed signal using the measure in Equation (2) for both the RTISI method and for G&L. The initial error of the G&L algorithm is large because of the zero phase initialization in all frames. Because RTISI uses the partially constructed frame as a starting point, RTISI produces a much smaller initial error. Furthermore, the algorithm converges quickly compared with G&L.taking only a few iterations before reaching a stable value. The error for G&L decreases monotonically and it typically becomes smaller than that of RTISI after about 15 iterations. After the point at which the error functions of the two methods cross, the reduction in error for further iteration of G&L is quite slow, and reasonable stopping criteria can be chosen to limit computational cost.

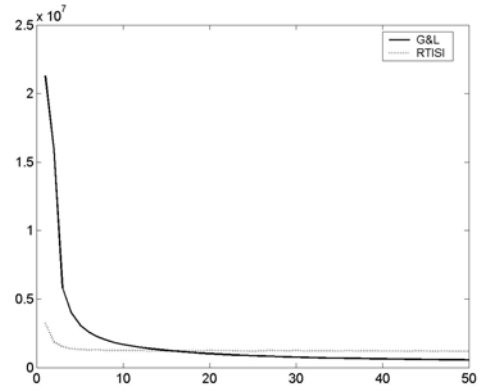

Figure 5. *Change of distance between the original signal and the reconstructed signal for G&L (solid line) and RTISI (dotted line).*

An obvious extension to the standard G&L algorithm for non-real-time applications is to run RTISI for one or two iterations to produce initial signal and phase estimates, and then proceed with G&L thereafter if a further reduction in error is still required. Comparing the tradeoffs in detail is part of our current research.

4.2. Time-Scale Modification Performance

For TSM applications, we applied the SOLA and RTISI algorithms to a variety of audio signal types including speech, music and generated signals such as chirp and pulse train. For a pulse train, which is the kind of signal SOLA was specifically designed to address, SOLA generates somewhat better results than RTISI. But for almost all other sound samples, SOLA does not perform as well as RTISI. The computation time for each iteration in RTISI is dominated by the fast Fourier Transform (FFT) and in each iteration for each N-sample frame we need to perform one N-point FFT and one N-point inverse FFT. For SOLA, each N-sample frame requires two 2N-point FFT's and one 2N-point inverse FFT. Generally RTISI achieves very good performance using 4 or 5 iterations, in which case the computational requirements for the two algorithms are similar.

When applied to polyphonic signals such as pop music, there are perceptually obvious artifacts with SOLA, even with a modification rate close to 1. Such artifacts include warbling, transient doubling and skipping, and tempo modulation. Informal listening tests show that RTISI works well for speech as well as for polyphonic signals such as music, even at modification rates far from 1. For TSM, there are no reference magnitude spectra to compare with the modified signal so we do not provide an objective evaluation function here, but the artifacts and distortions from RTISI are far less audible than those from SOLA. There is slight "phasiness" that can be perceived for simple nonstationary signals such as a chirp, similar to the problem addressed in [12] and [13] in the context of the vocoder, but for richer real audio signals the phasiness is not obvious.

Another advantage of RTISI over many time-domain TSM methods is that the algorithm makes no pitch estimates or periodic-versus-noise decisions. The straight-forward overlap-

and-add process of RTISI generates high quality results and the consistency between successive frames remains across a wide range of modification rates.

The most salient artifact produced by RTISI audio constructions is a smearing of transient information. The sub-frame sample-rate temporal information from the original signal is not used in the method, and the phase choices are made to enforce inter-frame consistency rather than any kind of transient pattern. Thus, there are less distracting artifacts like clicks and reverberation in RTISI than SOLA, but music loses some its "punch" and speech loses a small degree of articulation. The effect is most noticeable in highly transient signals such as click trains or the sound of a crackling fire. Examples are available at http://www.zwhome.org/~lonce/Publications/RTISI.html.

5. CONCLUSIONS

The Real-Time Iterative Spectrogram Inversion (RTISI) algorithm for constructing real audio signals from a sequence of magnitude spectra was presented. The method is based a strategy presented by Griffin and Lim [1], but is modified to estimate audio frames in sequence rather than in parallel. In addition to making the method applicable to real-time audio signal construction, the modification allows for better initial phase estimates which significantly speed up the convergence toward target spectra.

The RTISI method generally approaches, but not matches the performance of G&L run at 100 iterations in terms of magnitude spectrum error, however, the perceptual quality of RTISI after 5 iterations is quite good. If a magnitude spectrum error better than that which RTISI can deliver is necessary, the RTISI approach with one or two iterations can be used to produce an initial phase estimate for the G&L providing a significant speed improvement.

When applied to time-scale modification, RTISI compares favourably with the SOLA time-domain method of signal modification in terms of computational complexity, and favourably in terms of perceptual results on all but a few specially constructed signals. RTISI provides considerable improvement at both small and large scaling factors in terms of the absence of the clickiness, pitch and reverberation artifacts that plague time-domain methods of TSM.

Future work is necessary to quantify the frame transition artifacts that are not reflected in spectral SNR measures for this family of signal reconstruction techniques. The RTISI method will also be explored further in the context of other signal modification applications, and for improving the ability to capture and incorporate transient behavior from source signals in applications where such information is available.

6. REFERENCES

[1] D.W. Griffin, J.S. Lim, "Signal Estimation from Modified Short-Time Fourier Transform", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol.32, no. 2, Apr, 1984.

[2] S.H. Nawab, T.F. Quatieri, J.S. Lim, "Signal Reconstruction from Short-Time Fourier Transform Magnitude", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol.31, no. 4, Aug 1983.

[3] K. Achan, S.T. Roweis, B.J. Frey, "Probabilistic Inference of Speech Signals from Phaseless Sepctrograms", *Neural Information Processing Systems 16,* pp. 1393-1400*,* 2003.

[4] S. Roucos, A. M. Wilgus, "High Quality Time-Scale Modification for Speech", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol.10, pp. 493-496, Apr 1985.

[5] W. Verhelst, M. Roelands, "An Overlap-Add Technique Based on Waveform Similarity(WSOLA) for High Quality Time-Scale Modification of Speech", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol.2, pp. 27-30 Apr 1993.

[6] D. Dorran, R. Lawlor, E. Coyle, "High Quality Time-Scale Modification of Speech Using a Peak Alignment Overlap-Add Algorithm(PAOLA)", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol.1, pp. 6-10, Apr 2003.

[7] E. Moulines, F. Charpentier, "Pitch Synchronized Waveform Processing Techniques for Text-To-Speech Synthesis Using Diphones*", Speech Communication*, vol. 9, pp. 453-467, 1990.

[8] J. Laroche, M. Dolson, "Improved Phase Vocoder Time-Scale Modification of Audio", *IEEE Transactions on Speech, and Audio Processing*, vol. 7, No.3, May 1999.

[9] T.F. Quatieri, R.J. McAulay, "Shape Invariant Time-Scale and Pitch Modification of Speech", *IEEE Transactions on Signal Processing*, vol. 4, no. 3, Mar 1992.

[10] E. Moulines, J. Laroche, "Non parametric techniques for pitch-scale and time-scale modification of speech", *Speech Communication*, vol. 16, pp. 175-205, Feb 1995.

[11] M. Slaney, D.Naar, R.F. Lyon, "Auditory Model Inversion for Sound Separation", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 77-80, Apr 1994.

[12] M. Puckette, "Phase-locked Vocoder", *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics.,* 1995.

[13] J. Laroche, M. Dolson, "Phase-Vocoder: About this phasiness business", *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, 1997.