FAST IMPLEMENTATION FOR NON-LINEAR TIME-SCALING OF STEREO SIGNALS

Emmanuel Ravelli, Mark Sandler and Juan P. Bello

Centre for Digital Music, Dept. of Electronic Engineering
Queen Mary, University of London, UK
emmanuel.ravelli@elec.qmul.ac.uk

ABSTRACT

In this paper we present an improved implementation of Duxbury's adaptive phase-vocoder approach for audio time-stretching using non-linear time-scaling and temporal masked phase locking at transients [1]. We show that the previous algorithm has some limitations, notably its slow implementation and its incapacity to deal with stereo signals. We propose solutions to this problems including: an improved transient detection, a much faster implementation using the IFFT for re-synthesis and a method for stretching stereo signals without artifacts. Finally, we provide some graphical results and quantitative measures to illustrate our improvements.

1. INTRODUCTION

We can define time-stretching of audio signals as the process of changing the signal's temporal scale without modifying its frequency content. There are two main approaches to audio time scaling, namely: time-domain and time-frequency methods.Time-domain methods perform well on speech or monophonic signals. However results are not as satisfying when dealing with polyphonic signals. As an alternative, time-frequency methods are used for audio time-stretching. They are mostly based on the standard phase vocoder algorithm. Though the phase vocoder gives far better results for polyphonic signals, it also introduces some artifacts. The two main artifacts are known as *transient smearing* and *phasiness*. Transient smearing refers to the softening of attacks such that the natural percussiveness of the transient regions is lost. The reason for this is that the phase correction applied by the algorithm makes the assumption of a signal composed of nearly stationary sinusoids and thus is not valid during a transient. Phasiness, on the other hand, is often perceived as if a reverb-like effect has been added to the sound. This is due to a lack of phase coherence across frequency channels in a given frame.

In previous works, Duxbury [1] proposes a modification of the standard phase vocoder time-scaling algorithm that partially solves the problem of transient smearing and considerably reduces phasiness by means of phase-locking and onset-based transient preservation. However, for all its theoretical strengths, this algorithm suffers from a slow implementation and it is not optimized for time-stretching of stereo signals, constraining its usability in real-time real-world applications.

In this paper we propose a series of improvements that lead to a much faster implementation of Duxbury's algorithm. We also propose modifications that further reduce the transient smearing problem and make the method able to deal with stereo signals. The rest of this paper is organized as follows: in Section 2 we briefly describe the standard phase-vocoder approach; in Section 3 we outline Duxbury's method, explaining the principles of phase-locking and transient preservation; in Section 4 we introduce improvements to the approach and evaluate quantitatively the benefit of those improvements on the algorithm; finally, in Section 5 we present some conclusions and directions for the future.

2. STANDARD PHASE VOCODER APPROACH

This algorithm is composed by a three-stage process: analysis, transformation and resynthesis (see [2] for a detailed study).

Analysis stage: The STFT of the original signal x can be calculated as follows:

$$X\left(sR_a,k\right)=\sum_{n=0}^{N-1}h_a\left(n\right)x\left(n+sR_a\right)e^{-j\Omega_k n} \tag{1}$$

with the channel's center frequency $\Omega_k=\frac{2\pi k}{N}$ and $k=0,1,...,N-1$. h_a is the analysis window and R_a is the analysis hop size.

Transformation stage: The idea is to stretch the signal, by a scaling factor γ, by keeping the magnitude unchanged while modifying the phase in order to preserve the instantaneous frequency. An estimation of the instantaneous frequency is :

$$\widehat{\omega}\left(\left(s+1\right)R_a,k\right)=\frac{\Delta\varphi(\left(s+1\right)R_a,k)}{R_a} \tag{2}$$

where $\Delta\varphi(\left(s+1\right)R_a,k)$ is the unwrapped phase difference.The new synthesis phase can then be calculated as :

$$\psi\left((s+1)R_s,k\right)=\psi\left(sR_s,k\right)+R_s\widehat{\omega}\left(\left(s+1\right)R_a,k\right) \tag{3}$$

with the synthesis hop size $R_s=\gamma\times R_a$.

Synthesis stage: There are two implementations for the synthesis of the stretched signal. The first uses a bank of oscillators whose amplitude and frequencies vary over time. The second implementation uses the Inverse FFT and overlap-add. The latter is usually favored because it is much faster than the filter bank implementation. It can be described as follows: the transformation stage yields the synthesis FFT $Y\left(sR_s,k\right)$ that has the same magnitude as $X\left(sR_a,k\right)$ and the synthesis phase as calculated before. By calculating the IFFT of $Y\left(sR_s,k\right)$, we obtain short-time signals $y_s(n)$. These segments are then weighted by a synthesis window h_s and overlap-added to obtain the output signal $y(n)$.

$$y(n) = \sum_{s=-\infty}^{\infty}h_s\left(n-sR_s\right)y_s\left(n-sR_s\right) \tag{4}$$

$$y_s\left(n\right) = \frac{1}{N}\sum_{k=0}^{N-1}Y\left(sR_s,k\right)e^{j\Omega_k n} \tag{5}$$

3. ADAPTIVE PHASE VOCODER APPROACH

It is generally accepted that at transients, artifacts may be perceptually "buried" [3], therefore we can sacrifice accurate frequency content in order to lock the synthesis phase at transients without any meaningful perceptual effect. The use of this idea for eliminating transient smearing is at the heart of the adaptive phase-vocoder approach to time-stretching proposed in [1]. In this approach, which is described briefly in the following, the signal is segmented using an onset detection algorithm, such that only the steady-state sections are time-scaled. Percussive transients are then selected within the signal as candidates for the masked phase locking. The analysis-transformation process can be divided into three main stages : pre-analysis, non-linear phase vocoding and rhythm preservation, and temporally-masked phase-locking at transients.

Pre-Analysis: onsets in the signal are located using a fixed resolution subband analysis and a complex domain detection function. A constant-Q filterbank splits the signal into four subbands. A complex domain detection function is calculated separately for each subband. After peak picking and thresholding each subband detection function, we obtain a range of detected onsets. The onsets are then combined in order to keep a maximum of one onset for a short window of 50ms. As the higher frequency subbands have better time resolution, a higher band always takes precedence over a lower band, therefore maximizing the accuracy of results.
A simple rule is used to differentiate tonal from percussive onsets: tonal onsets are described as being detected in only one subband within a 20 ms temporal window, while percussive onsets are detected in at least two subbands. Detected onsets divide the signal in temporal segments corresponding to "notes". Each "note" is then splitted into a transient part (just after the onset) and a steady-state part (the rest). The width of the transient is assigned as 1/3 of the time until the next onset, up to a maximum width of 50 ms and a minimum width of 11.6 ms.

Non-linear phase vocoding and rhythm preservation: only the steady-state parts of the signal are stretched, the transients regions are shifted in time, with their width preserved. In order to preserve rhythm, a variable stretching factor F must be calculated for each steady-state region. If γ is the global stretching ratio, F can be calculated directly using :

$$F = \frac{W_{nts}}{W_{ts}}(\gamma - 1) + \gamma \tag{6}$$

where W_{ts} is the width of the time-scaled region (steady state) and W_{nts} is the width of the non time-scaled region (transient).

Temporally masked phase locking at transients: in order to solve the problem of transient smearing, this algorithm locks the synthesis phase to the analysis phase in the temporally masked region of the percussive transients. This has the advantage that, for many signals, the phase error does not propagate across enough frames to be considerable before the phase is locked again. This approach is especially useful when stretching signals which comprise both steady-state and percussive attacks, for which many other methods produce poor results. The phase locking is done using the previous synthesis phase $\psi(sR_a, k)$ in place of the previous analysis phase $\varphi(sR_a, k)$ in the unwrapped phase difference calculation (Eq. 3). This value is then used in the first frame (where the signal is not stretched) of a percussive transient region to lock the phase, thus preserving the attack transient.

4. AN IMPROVED IMPLEMENTATION AND ITS EVALUATION

Though the previous approach gives very good results compared to the standard algorithm, it has several problems and limitations. At first, the used subband onset detection algorithm is successful at detecting onsets, but it often suffers from problems of localization, preventing the total cancellation of the transient smearing artifact. A solution to this is proposed in 4.1. Next, Duxbury's algorithm uses the oscillator-bank implementation of the phase vocoder for the re-synthesis stage. This is very slow and not feasible for real-time applications. To reduce the computational cost, we have to use the implementation with the IFFT and overlap-add. But this implementation has some problems with non-linear time scaling that the oscillator bank implementation has not. This is discussed in 4.2. Finally, Duxbury's algorithm deals with mono channel signals and introduces some artifacts when stretching stereo signals. The reason for this is that the pre-analysis is performed independently on each channel resulting in a lack of phase coherence between the channels. A proposed solution is discussed in 4.3.

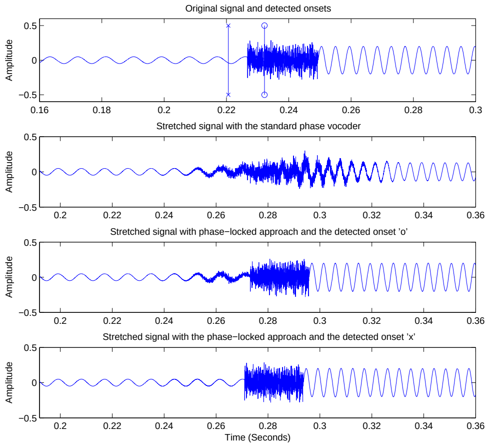

Figure 1: *Illustration of the transient smearing artifact*

4.1. Reducing transient smearing

Good performance in [1] relies on the accuracy of the onset detection. However, his detection algorithm occasionally suffers from problems of localization. The positions of detected onsets are often delayed against those of real onsets. Consequently, a small section of the transient is missed, resulting in transient smearing.We propose a simple solution to this problem that uses the first derivative of the complex domain detection function. The obtained onsets are consistently better located, and as a consequence the phase is locked just before the transient, keeping the transient intact and reducing the transient smearing effect to virtually zero.
To show that the first derivative of the complex domain function is better for our time-scaling algorithm, we construct a signal that simulates a percussive onset. This signal is composed of 10000 samples of a 110 Hz sinusoid (simulating steady-state) followed by 1000 samples of white gaussian noise (simulating a transient), which are in turn followed by 10000 samples of a 220 Hz sinusoid

(simulating the steady-state of the new note). At the top of figure 1 the original test signal can be observed, followed below by the stretched signal ($\gamma = 1.2$) using the standard phase-vocoder technique, clearly showing the transient smearing artifact. It can be noted how the onset detected using the complex domain detection function (marked with an 'o' in the top plot) is within the transient, thus generating transient smearing in the stretched signal (third plot from top). The onset detected using the first derivative of the complex domain detection function (marked with an 'x') provides better segmentation of the signal, thus eliminating the transient smearing artifact in the stretched signal (bottom plot).

4.2. Improved speed with an IFFT implementation

As we want to implement the algorithm in a real time-application, synthesis using IFFT and OLA needs to be considered. However, this implementation presents some problems with nonlinear time-scaling. Using overlap-add (OLA), the envelope of the synthesis signal is normalized only when the sum of the products of analysis and synthesis windows equals unity. Indeed, using a variable stretching factor induces a variable synthesis hop size (the analysis hop size remains constant), and consequently, the resulting envelope of the OLA process is not normalized, introducing a vibrato-like effect (see solid-line windows and envelope in figure 2). There are several methods for solving this problem [4], each showing advantages and disadvantages.

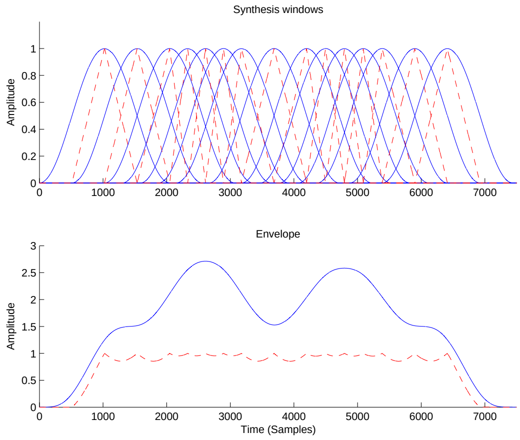

Figure 2: *Synthesis windows and resulting envelopes*

The optimal solution is to post-normalize the synthesis signal using the calculated envelope. Though this solution is much faster than the filterbank implementation, this is only possible once the signal has been completely synthesized, so it is inappropriate for frame-by-frame processing. Another possible solution is to obtain γ by using a variable analysis hop size and a constant synthesis hop size. The synthesis signal could then be normalized (if the synthesis hop size is a divisor of $N/4$ [2]) and frame-by-frame processing is possible. However, it is inappropriate in our case because a variable analysis hop size implies that the phase-vocoder analysis is not synchronized with the onset detection algorithm resulting in poor localization of the phase-locked frames. A compromise solution is to use asymmetric triangular windows for the synthesis instead of using hanning windows (see dashed windows and envelope in figure 2). The envelope is then approximately normalized and a frame-by-frame processing is possible. The choice of synthesis method depends on the application. If it is for an offline processing, then the first method is the best. If it is for a frame-by-frame processing, then one has to choose the third method. Table 1 shows processing times for 120% stretching using the oscillator bank synthesis and using the first or the third IFFT synthesis method. The implementation is made with Matlab on a pentium M at 1.6 Ghz. Results clearly show that the synthesis with the IFFT is more than 30 times faster than the filter-bank implementation.

	Oscillator bank	**IFFT+OLA**
Test signal (length : 0.5 s)	9 s	0.3 s
Pop (length : 6 s)	223 s	7 s
Electronic (length : 9 s)	314 s	10 s

Table 1: Comparison of processing times

Because fast implementations come at a cost in the quality of the synthesis signal, we now have to quantitatively evaluate the impact of using an IFFT-based approach to re-synthesis. It is very difficult to establish a solid measure for the performance evaluation of time-scaling algorithms. A measure of "phasiness" is proposed in [5], based on the fact that the STFT of the synthesized signal is close to the transformed spectrum $Y(sR_s, k)$ in both amplitude and phase if the time-scaling algorithm ensures phase coherence. Therefore a "Distance Measure" can be defined as:

$$D_M = \frac{\sum_{s=1}^{s=N_f} \sum_{k=0}^{N-1} \left|Z\left(sR_s, k\right) - Y(sR_s, k)\right|^2}{\sum_{s=1}^{s=N_f} \sum_{k=0}^{N-1} \left|Y(sR_s, k)\right|^2} \qquad (7)$$

with $Z\left(sR_s, k\right)$ the STFT of the synthesized signal. The smaller D_M, the lesser the phasiness in the synthesized signal.
The phasiness distances for different signals and different algorithms are given in Table 2. The algorithms are: the standard phase vocoder; Duxbury's adaptive phase vocoder; our fast implementation with improved onset detection, IFFT and post-normalization for the OLA of the synthesis signal; same but using asymmetric triangular windows for the OLA.

	Test	**Pop**	**Music**	**Electro**	**Music**
	Signal	Left	Right	Left	Right
Standard	-2.6	-2.1	-3.0	-2.8	-2.5
Duxbury	-20.8	-5.2	-4.9	-7.4	-7.5
IFFT post	-22.9	-5.7	-5.4	-8.2	-8.8
IFFT asym	-15.7	-5.3	-5.1	-7.5	-8.1

Table 2: Distance measures for stretched signals (in dB)

These results clearly show that the phase-locked approaches considerably reduce phasiness when compared to the standard approach. The results obtained with the IFFT approaches are slightly better than those obtained with the oscillator bank implementation, with the added benefit of a much faster implementation. Best results are obtained with the the post-normalized approach to OLA, however, the results of the asymmetric windows synthesis approach remain comparable. Informal listening tests confirms results in the table.

4.3. Time-stretching stereo signals

Duxbury's approach performs well on mono channel signals but introduces artifacts when stretching stereo signals. The pre-analysis is performed independently on each channel, yielding an independent set of tonal and percussive onsets for each. Consequently, if the transient detection scheme is applied incorrectly on one of the two channels or if the signal is strongly panned, we loose the phase coherence between the channels, resulting in the sound "moving" rapidly between channels. The reason for this is that the rhythm preservation, as explained in 3, is not perfect as the real stretching factor F is not equal to the theoretical one because the synthesis hop size is necessarily a round integer. Consequently, a little shift is introduced for each detected onset, loosing the phase coherence between channels. The solution is to forcibly detect single-channel onsets in both channels. Then, we compute the cross-correlation between channels around each detected onset. We assume that if the two channels are locally similar (i.e. the maximum of the cross-correlation is at the middle), the pre-analysis algorithm must detect the same percussive onset in both channels. If they are not, we "force" this detection to occur : we change a tonal onset into a percussive onset if there is a corresponding percussive onset on the opposite channel. This re-classification introduces a few artifacts in the independent signals, but the quality of the overall stretched stereo signal is greatly improved by the forced coherence.

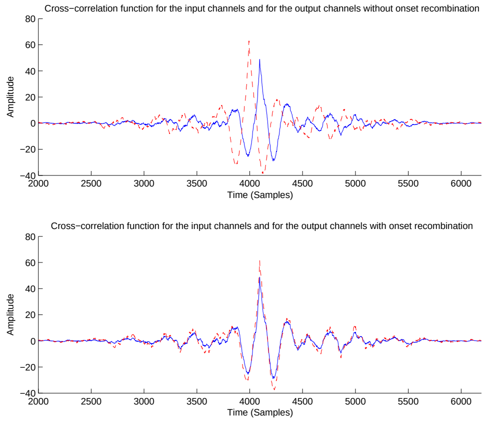

Figure 3: *Cross-correlation for one frame*

To show this improvement, we propose a measure of the phase coherence between the two channels of a stretched stereo signal as:

$$C_M = \frac{\sum_{s=1}^{s=N_f} \sum_{m=0}^{L} |C_{y_1 y_2}(sR_s, m) - C_{x_1 x_2}(sR_a, m)|^2}{\sum_{s=1}^{s=N_f} \sum_{m=0}^{L} |C_{x_1 x_2}(sR_a, m)|^2} \quad (8)$$

with $C_{x_1 x_2}$ the short-time cross-correlation between channels of the original signal and $C_{y_1 y_2}$ the short-time cross-correlation between channels of the stretched signal. This measure is based on the fact that the cross-correlation of the input channels and the cross-correlation of the output channels must be comparable, i.e. the distance between correlations is small. If not, it means that the time-stretching algorithm introduce phase incoherence between the two channels, resulting in misaligned cross-correlation functions and thus a larger distance value, as seen in figure 3.

Table 3 shows C_M for stretched polyphonic stereo signals, both without and with "forced" onset synchronization. The results clearly show that our solution reduces the phase incoherence between channels. Again, informal listening tests confirm that the stereo artifact disappears with this improvement.

	Without improv.	**With improv.**
Pop	16.6 dB	15.2 dB
Electronic	18.1 dB	14.2 dB
Pop2	16.7 dB	14.3 dB
Electronic2	24.0 dB	22.5 dB

Table 3: Measure of the phase coherence between channels

5. CONCLUSIONS AND NEXT STEPS

Duxbury's modification of the standard phase vocoder approach to time-scaling partially solves the problem of transient smearing and reduces phasiness, however this implementation suffers from several limitations with regards to computational cost and the handling of stereo signals. First, we propose a simple way of improving the accuracy of the onset detection, resulting on an improved cancellation of the transient smearing. We also propose a much faster implementation which is suitable for real-time applications and that shows no loss of quality, as clearly demonstrated by our experiments. Finally, the case of stereo signals has been considered and the problem of phase incoherence between channels has been solved by a suitable recombination of the onsets.
The general framework of the time-scaling algorithm presented here could easily form the basis of a more general time-modification tool. As we have the location of both the onsets and transients, it could also be used in more diverse applications. We are currently studying the temporal re-arrangement of events in recorded signals for rhythm modification tasks.

6. ACKNOWLEDGMENTS

The authors wish to thank Dr. Chris Duxbury for making his code available and for his insights regarding the problems at hand. This work was partially funded by the European Commission through the SIMAC project IST-FP6-507142.

7. REFERENCES

[1] Christopher Duxbury. *Signal Models for Polyphonic Music*. PhD Thesis, 2004.

[2] D. Arfib, F. Keiler, and U. Zolzer. *Time-frequency Processing*, chapter in DAFX: Digital Audio Effects. Wiley, 2002.

[3] B.C.J. Moore. *An Introduction to the Psychology of Hearing*. Academic Press, fourth edition, 1997.

[4] V. Verfaille. *Effets audionumeriques adaptatifs : theorie, mise en oeuvre et usage en creation musicale numerique*. PhD Thesis, 2003.

[5] J. Laroche and M. Dolson, "Improved Phase Vocoder Time-Scale Modification of Audio," *IEEE Trans. Speech Audio Processing*, vol. 7, no 3, pp. 323-332, May. 1999.