Audio Engineering Society

Convention Paper

Presented at the 142nd Convention

2017 May 20–23 Berlin, Germany

# Audio Time Stretching with Controllable Phase Coherence

Nicolas Juillerat[1]

[1] *University of Fribourg, Switzerland*

Correspondence should be addressed to Nicolas Juillerat (nicolas.juillerat@unifr.ch)

**ABSTRACT**

This paper presents a hybrid audio time stretching technique in which the trade-off between vertical and horizontal phase coherence can be freely controlled by a single parameter. Depending on that parameter, the proposed technique sounds like a time domain technique at one extreme, like a phase-locked vocoder at the other extreme, or anywhere in between. By properly choosing the value of the control parameter, it is possible to manually adjust the algorithm to the characteristics of the audio signal being transformed in order to get an optimal result. Furthermore, appropriate middle values yield good results for a wide range of audio signals with mixed content.

## 1 Introduction

Most audio time stretching implementations fall into one of the following two categories: time domain techniques, and phase vocoder-based techniques [1].

Time domain techniques work well for voice and for small changes. However, the quality of the transformation quickly degrades when a large change is applied, especially with polyphonic material, where warbling and stuttering artefacts are typically heard.

The phase vocoder on the other hand can accommodate both polyphonic music and large changes, but suffers from *phasiness*, especially on speech and vocals that tend to sound “distant” or “reverberant” [2].

The phase vocoder works in the frequency domain, and adjusts the phase of every frequency independently to ensure continuity over time; it preserves what is known as *horizontal phase coherence*.

Time domain techniques on the other hand preserve the phase relations between all frequencies at every point in time; this is *vertical phase coherence*.

Note that both approaches try to some degree to preserve both vertical and horizontal phase coherence, but only one is “fully” preserved, while the other one is only preserved as much as possible.

Improved phase vocoder approaches for instance use phase locking around peaks in the spectrum to preserve some vertical phase coherence [3]. Improved time domain approaches use advanced similarity measures between short-time blocks to preserve some horizontal phase coherence [4].

The technique proposed in this paper allows one to arbitrarily choose by how much horizontal phase coherence is favored at the detriment of vertical phase coherence, or vice versa.

Hence it is possible to freely adapt the phase coherence to the nature of the audio being processed. For example, speech and vocals can be processed with more vertical phase coherence to prevent phasiness, while symphonic music or high stretching ratios may benefit from more horizontal phase coherence. Furthermore, choosing an intermediate trade-off that does not favor any kind of phase coherence gives good results on a wide range of audio signals with mixed content.

This paper is organized as follows. Section 2 discusses related work. Section 3 summarizes the phase-locked vocoder on which the proposed technique is based. The proposed technique is described in terms of successive improvements in Sections 4 to 7. Section 8 validates the approach by measuring the horizontal and vertical phase coherence and comparing it to other techniques. Further improvements are highlighted in Section 9.

## 2 Related Work

Several techniques have been proposed to improve the vertical phase coherence of the phase vocoder or to improve the horizontal phase coherence of time domain techniques.

- The PVSOLA algorithm [5] uses a time domain technique to regularly reset the phases of a phase vocoder, to get a relatively high, but not freely controllable vertical phase coherence.
- Phase changes that are below or close to the threshold of audibility can be used to improve the vertical phase coherence of the phase vocoder [6]. While some control is possible, this technique is limited to small time stretching ratios.
- Shape invariant time stretching fully preserves vertical phase coherence [7]. However, it is difficult to implement on polyphonic or inharmonic material.
- The horizontal phase coherence of time domain techniques can be increased by applying the transform independently on several frequency bands [8]. The phase coherence is then directly controlled by the number of bands. However, the computation time increases proportionally to that number.

The proposed technique allows the amount of vertical phase coherence to be freely controlled, and has the advantage of working at any time stretching ratio and with any audio signal. Furthermore, the computation time remains constant.

## 3 The Phase-Locked Vocoder

The presented technique is essentially based on the phase-locked vocoder by Laroche and Dolson [3], with several modifications. These modifications can also be applied on the more recent phase-locked vocoder proposed by Bonada [9].

The phase-locked vocoder uses the short-time Fourier transform (STFT). The STFT is defined by the length of the Fourier transform $N$, and by the analysis and synthesis hop sizes $R_a$ and $R_s$. The time stretching ratio $\alpha$ is defined by $\alpha = \frac{R_s}{R_a}$. An analysis window $W(t)$ is used to multiply the signal before the Fourier transform, and a synthesis window $V(t)$ to multiply it again after the backward Fourier transform. Given an audio signal $x(t)$ in the time domain, the STFT transforms it into a spectrum given by $X[s,k]$, where $s$ is the frame index. It is related to time by $t = sR_a$. $X[s,k]$ is a *bin*, a complex number that gives the phase and magnitude of the signal at the frame index $s$ and at the normalized frequency $\frac{k}{N}$. For a given frame index $s$, the list of values $X[s,k]$ for $k \in [0, \frac{N}{2}]$ is a *spectral frame*. The overlapping factor is defined as $\frac{N}{R_s}$. Typical values range from 2 to 8.

The way a spectral frame is processed by the phase-locked vocoder can be summarized as follows:

- Peaks are identified in the spectral frame. A bin is a peak if its magnitude is greater than that of its four nearest neighbors. A typical polyphonic music might contain about 300 to 500 peaks for a typical block size $N$ of 4096 at 44.1 kHz.
- Phase propagation is performed on every peak. For the details of this process, the reader can refer to previous work by Laroche and Dolson [3]. It suffices to say that, for each peak $p$ of a spectral frame $s$, this process results in a phase coefficient $\varphi_{s,p}$ that is added to the phase of the corresponding peak bin.

- Phase locking: the phase of every non-peak bin is modified by adding $\varphi_{s,p}$, where $p$ corresponds to the closest peak. This corresponds to the "identity" phase locking scheme. Other phase locking schemes are possible [3].

The phase locking step is essentially motivated by the following assumptions:

- A single sinusoid, unless it is perfectly centered on a bin, results in several non-zero bins, as given by the main lobes and side lobes of the analysis window.
- A chirp signal results in an "enlarged" main lobe compared to a sinusoid.

In both cases, a single signal results in many bins around a center one, and they all correspond to the same frequency. Hence it makes sense to process them in the same way.

## 4 Inter-Peak Locking

The presented technique extends the phase-locked vocoder by not only locking the phase around peaks, but also *between* peaks. Two questions must be answered:

- How to choose the groups of peaks that are locked together?
- How to lock the phase between peaks of the same group?

### 4.1 Grouping the peaks

To choose the groups of peaks, a scheme that is both simple and effective is to divide the spectral frames into $M$ frequency bands on a logarithmical scale, and to group the peaks that are in the same band. Another possibility would be to group the peaks that correspond to a fundamental frequency and its harmonics, but this would be harder to implement, and would not handle inharmonic content.

The current implementation uses the first approach, with a modified logarithmical scale in which bands are forced to span at least 3 bins. This makes sense for two reasons. First because a band can only contain two peaks or more if it spans more than 3 bins, so there is no reason to go below 3. Second, this gives a well-defined maximum value for $M$ of $\frac{N+1}{6}$. The number of bands $M$ allows one to control the trade-off between horizontal and vertical phase coherence:

- A value of $\frac{N+1}{6}$ (the maximum) results in every band to contain zero or one peak. As such there is no locking between any peaks and the processing matches the phase-locked vocoder.
- A value of one (the minimum) brings all the peaks in the same single band, and results in a "global" locking. This is similar to time domain techniques that consider audio chunks in time only, without any frequency subdivision.
- Intermediate values give intermediate trade-offs, with smaller values favoring vertical phase coherence and larger values favoring horizontal phase coherence.

### 4.2 Locking the phase between peaks

For each group of peaks, a main peak $p$ is chosen, for example the peak that has the highest magnitude. Phase propagation is performed as usual on that peak using $\varphi_{s,p}$ as discussed in Section 3.

For other peaks $p_k$ of the same group, the phase propagation $\varphi_{s,p_k}$ is explicitly defined based on that of the main peak $p$:

$$\varphi_{s,p_k} = \frac{\varphi_{s,p}\Omega_{s,p_k}}{\Omega_{s,p}} \qquad (1)$$

Where $\Omega_{s,p}$ is the estimated frequency of peak $p$ at frame index $s$.

Once $\varphi_{s,p_k}$ is determined for all other peaks of the group, the usual phase locking of the phase-locked vocoder is used for bins *around* each peak.

The idea behind Equation (1) is to lock to the "time shift" given by $\frac{\varphi_{s,p}}{\Omega_{s,p}}$ rather than to the phase $\varphi_{s,p}$. This better matches time domain techniques that are based on shifting blocks in time.

Furthermore, Equation (1) is essentially a generalization of phase locking as used in the phase-locked vocoder. Indeed, when locking is only done around a single peak, the frequencies $\Omega_{s,p}$ and $\Omega_{s,p_k}$ can be dropped from Equation (1) because they are precisely assumed to be equal. It then reduces to $\varphi_{s,p_k} = \varphi_{s,p}$ , which corresponds to the identity phase locking of the phase-locked vocoder.

## 5 Using Multiple Overlaps

In general, an overlap of 2 (50%) is considered low quality for the phase-locked vocoder, and better results are achieved using an overlap of 4 or greater. However, when locking peaks together using Equation (1), increasing the overlap substantially *reduces* the quality of the result, especially with small values of $M$. This is indeed expected: the smaller the value of $M$, the lower the horizontal phase coherence, and the less coherent the addition of overlapped STFT frames. Perceptually, the result sounds more "metallic" and objectionable when the overlap is increased.

This problem is solved by:

- Using a first STFT with a high overlap (4 or greater) for the low frequency bands that span 3 bins or less, because none of them can possibly contain more than one peak. These bands are hence processed like an unmodified phase-locked vocoder.
- Using a second STFT with an overlap of 2 for other frequency bands that can potentially contain 2 peaks or more. These bands are processed using the inter-peak locking discussed in Section 4.
- Coping directly with some problems that occur when an overlap of 2 is used, namely with noise. This is discussed in the next two sections.

These choices are further motivated by the fact that time domain techniques generally use a low overlap as well, sometimes even less than 2.

## 6 Handling Noise

When an overlap of 2 is used, the analysis window $W(t)$ and synthesis windows $V(t)$ do not usually add to a constant. This is not a problem in practice as it suffices to compute the resulting curve $D_T(t)$ using Equation (2) and to use it to divide the time domain signal after the STFT.

$$D_T(t) = \sum_s W(t + sR_s)V(t + sR_s) \qquad (2)$$

The problem is that Equation (2) assumes that the windows add *coherently*. This is however only true for tonal signals for which horizontal phase coherence can be preserved. For noise and with an overlap of 2, unless the time stretching ratio $\alpha$ is close to 1, the windows add *incoherently*:

$$D_N(t) = \sqrt{\sum_s W(t + sR_s)^2 V(t + sR_s)^2} \quad (3)$$

In general, $D_T(t) \neq D_N(t)$.

When the overlap is significantly greater than 2, Equation (2) can generally be used for both tonal and noise components, at least when an advanced phase locking scheme such as scaled phase locking [3] is used. The reason is that a high overlap introduces correlation in time between STFT bins, which makes noise add more coherently [1]. The introduced correlation gets lost in the overlap-add process of the backward STFT.

Hence the process is further enhanced as follows to cope with noise:

- A noise detection algorithm is used to classify every STFT bin as either "noise" or "tonal".
- For the backward STFT with high overlap (for low frequencies), $D_T(t)$ is used to divide the resulting time domain signal, regardless of the classification of the STFT bins.
- The backward STFT with an overlap of 2 is further split in two: one instance for tonal components, using $D_T(t)$ to divide the resulting time domain signal; and one instance for noise components, using $D_N(t)$. Note that two instances must be used because it is generally not possible to find a constant $c$ such that $D_N(t) = cD_T(t)$ holds when an overlap of 2 is used.

[1] In fact, some correlation is introduced even with an overlap of 2, but is generally negligible with commonly used analysis windows.

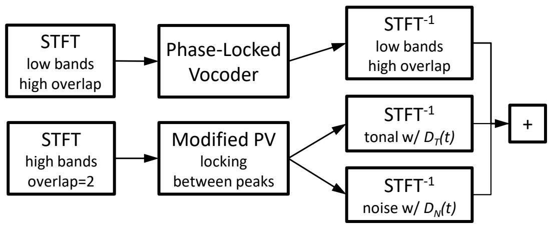

Figure 1: Forward and backward STFTs involved

In summary, there are *two* forward STFTs, and *three* backward STFTs, as shown in Figure 1. In practice, the forward STFT with an overlap of 2 can be implemented trivially by dropping spectral frames from the forward STFT of high overlap. This allows the implementation to use a single forward STFT and hence also a single noise detection step.

Without any proper handling, noisy components of a transformed music are perceived as "beating", especially with small values of $M$. This effect is mostly mitigated (limited to the accuracy of noise detection) using the technique described in this section. Samples are available online [10].

## 7 Detecting Noise

To implement the process discussed in the previous section, it is necessary to detect STFT bins that consist of noise. Several approaches have been proposed in the literature [11], [12], [13].

The following simple and effective criterion inspired from previous work [12] has been used:

$$|X[s,k]|^2 < \delta \underset{m=k-u(k)/2}{\overset{k+u(k)/2}{median}} (|X[s,m]|^2) \tag{4}$$

A simple and effective choice for $u(k)$ is to choose a constant corresponding to a fixed bandwidth somewhere between 100 and 500 Hz. Another reasonable choice is to match the bandwidths of the Bark bands. The $\delta$ factor gives good results when set to about 10.

Equation (4) alone is not enough. When the time stretching ratio $\alpha$ is close to 1, even noise adds itself mostly coherently. To detect that situation, the following criterion has experimentally shown to work well in practice (using unwrapped phases):

$$|\varphi_{s,k} - \varphi_{s-1,k}| \le 2\pi\sigma \tag{5}$$

The constant $\sigma$ is a "coherence" threshold substantially less than 1. Indeed, noise is expected to add mostly coherently when the left hand-side of Equation (5) is confined to values sufficiently close to zero to be insignificant. Values of $\sigma$ between 0.1 and 0.3 have experimentally shown to give good results in practice.

For performance reasons, it is preferable not to have to compute the phase shifts $\varphi_{s,k} - \varphi_{s-1,k}$ for all the bins. To derive a faster and more stable criterion, observe that the purpose of the phase propagation step of the phase-locked vocoder is to compensate for the time shift between frames introduced by using different analysis and synthesis hop sizes. This time shift is given by $R_a(\alpha - 1)$. The corresponding phase shift can be approximated by multiplying this expression by $2\pi\Omega_{s,k}$, where $\Omega_{s,k}$ is the frequency of bin $k$ at frame index $s$. By using the approximation $\Omega_{s,k} \approx \frac{k}{N}$, and dropping the common factor $2\pi$, the following faster criterion is obtained:

$$|\frac{k}{N} R_a(\alpha - 1)| \le \sigma \tag{6}$$

A last criterion can be added. When not in the presence of noise, it may still happen that the inter-peak locking of Equation (1) results in a large phase difference $|\varphi_{s,p} - \varphi_{s-1,p}|$ for some peaks. If this difference is too large, horizontal phase coherence will not be preserved for the corresponding peak. This might occur when a small number of bands $M$ are used, and when a peak is locked to another peak of a substantially different frequency. The following additional criterion can be used:

$$|\varphi_{s,p} - \varphi_{s-1,p}| > 2\pi\sigma \tag{7}$$

If Equation (7) is verified, the peak $p$ and its surrounding bins are treated like noise because they do not preserve horizontal phase coherence, and will hence add incoherently.

As a summary, the full criterion for selecting bins requiring incoherent addition is given by:

$$((4)\ AND\ NOT\ (6))\ OR\ (7) \tag{8}$$

Note that only Equation (4) actually "detects" noise, Equations (6) and (7) correspond to the two special cases in which noise adds coherently and tonal components add incoherently, respectively.

It should be noted that all criteria are based on a binary tonal / noise decomposition, and make use of various experimental constants. As such, additional improvements might be considered for future works.

## 8 Validation

The presented algorithm has been implemented and validated by comparing it against the unmodified phase-locked vocoder [3] and the WSOLA (Window Similarity Overlap Add) time domain technique [4]. This was done by transforming test signals, and approximating both the horizontal and vertical phase coherence of the results. A time stretch ratio of $\alpha = 1.5$ was used, with different values of $M$ for the presented technique. The size of the STFT was $N = 4096$ at 44.1 kHz, which gives a maximum value of $\frac{N+1}{6} = 683$ for $M$.

### 8.1 Horizontal phase coherence measure

When horizontal phase coherence is *not* preserved, consecutive STFT frames add incoherently and the whole process *reduces* the amplitude of the resulting signal if this is not compensated for.

The amplitude of a signal $x(t)$ can be aggregated into a single value $\mathcal{A}(x)$ using the root mean square ($L$ is the length of the signal $x(t)$ in samples):

$$\mathcal{A}(x) = \sqrt{\frac{1}{L}\sum_{t=0}^{L-1} x(t)^2} \tag{9}$$

Then, the horizontal phase coherence $\mathcal{H}$ is approximated using the following simple, yet effective equation, where $y(t)$ is the resulting time stretched signal:

$$\mathcal{H} = \frac{\mathcal{A}(y)}{\mathcal{A}(x)} \tag{10}$$

| ***M*** | **1** | **10** | **40** | **100** | **250** | **683** |
|---|---|---|---|---|---|---|
| (a) | -2.15 | -1.77 | -0.73 | -0.54 | -0.50 | -0.48 |
| (b) | -0.33 | -0.30 | -0.40 | -0.44 | -0.47 | -0.48 |
| (c) | -0.27 | -0.25 | -0.20 | -0.10 | -0.06 | -0.03 |

Table 1: Estimation of the horizontal phase coherence $\mathcal{H}$ (in dB).

The better the horizontal phase coherence is preserved, the closer $\mathcal{H}$ is to 1 (or 0 dB).

Table 1 shows the values of $\mathcal{H}$ measured and averaged over different test signals comprising classical, pop, latin and electronic music for different values of $M$. The values of $\mathcal{H}$ are given in decibels.

Row (a) is the result when locking the phase across peaks, and using an overlap of 8 (Section 4). The loss of horizontal phase coherence is evident as $M$ gets smaller. Perceptually, the transformed audio sounds "thinner" and more "metallic", and is clearly objectionable. This motivates the use of multiple overlaps.

Row (b) is the result when multiple overlaps (2 and 8) are used (Section 5): the measured horizontal phase coherence is much less varying with $M$. It is in fact slightly decreasing as $M$ increases. This suggests that using a low overlap is better in preserving horizontal phase coherence than the phase propagation of the phase vocoder with a high overlap. This can be explained by the fact that most of the loss of horizontal phase coherence occur during the phase propagation of *noise* components. Perceptually though, lack of horizontal phase coherence on noise is hardly audible, a fact that is not taken into account by Equation (10).

Row (c) is the final result, when also handling noise (Section 6). Observe that the measured values are the closest to 0 dB, which would correspond to an optimal result according to the proposed measure. Note that strictly speaking, noise handling does not *actually* improve the horizontal phase coherence, but rather compensate for the amplitude attenuation of noise components that results from their lack of horizontal phase coherence.

As a comparison, the value of $\mathcal{H}$ was measured and averaged on the same test signals as -0.48 for an unmodified phase-locked vocoder, and as -0.11 for time domain WSOLA. Table 1 shows that when $M$ is set to the maximum, noise handling gives an improvement (-0.03) over the unmodified phase-locked vocoder. On the other hand, with the minimum value of $M$, the presented algorithm (-0.27) is still slightly below WSOLA. This can be explained by the fact that a single peak is used to determine a common time shift, whereas WSOLA uses a more elaborate similarity measure based on correlation. Implementation details such as the chosen frequency resolution, or frequency domain versus time domain processing might also come into play.

## 8.2 Vertical phase coherence measure

Measuring the vertical phase coherence is harder, especially for arbitrary signals. Hence it was measured using musical signals to which vertical phase coherence was *artificially added*.

To artificially add vertical phase coherence to a signal, an STFT is performed with rectangular windows and an overlap of 1 (no overlap). The phase of every STFT bin is just set to zero. This zero-phasing transformation makes the modified signal look like a pulse train, as shown by Figure 2 (b).

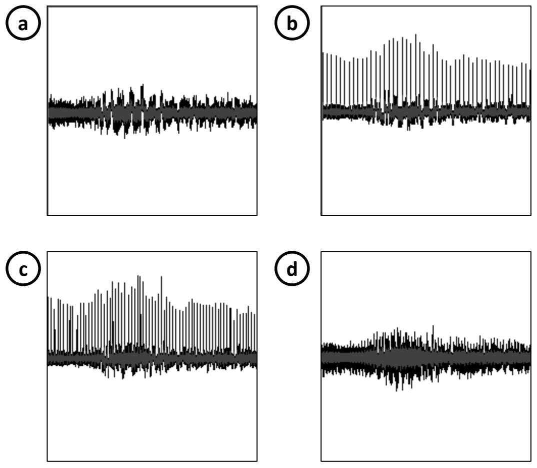

Figure 2: (a) Musical signal, (b) zero phased, time stretched by 1.5 with (c) good and (d) poor vertical phase coherence.

The size of the STFT was chosen to get a period of about 13.3 ms, to get pulses at 75 Hz. Such a frequency is sufficiently low so that it is mostly heard as distinct pulses (although very fast), yet it is sufficiently high so that every STFT frame contains several pulses, when the time stretching is done with a typical value of $N = 4096$ at 44.1 kHz.

A true synthesized pulse train cannot be used because it would entirely consist of stationary sinusoids that are multiples of the base frequency. A signal only consisting of stationary sinusoids is a special case in which the phase-locked vocoder fully preserves the vertical phase coherence [3]. However, such signals are extremely rare in practice. Using the zero-phasing transformation on real musical signals, enough characteristics from the original signals are preserved so that the resulting test signal does not fall in that special case.

The zero phased test signals have a higher crest factor (maximum divided by root mean square) than the original signals, as illustrated by Figure 2 (a) and (b). If vertical phase coherence is *not* preserved, the time stretching process is expected to *reduce* the crest factor. An averaged crest factor $\mathcal{C}(x)$ can be estimated using short-time contiguous blocks of $N$ samples:

$$\mathcal{C}(x) = \frac{N}{L} \sum_{s=0}^{L/_{N}-1} \frac{max_{i=0}^{N-1} x(sN+i)}{\sqrt{\frac{1}{N} \sum_{i=0}^{N-1} x(sN+i)^2}} \qquad (11)$$

Then, the vertical phase coherence $\mathcal{V}$ is approximated as follows:

$$\mathcal{V} = \frac{\mathcal{C}(y)}{\mathcal{C}(x)} \qquad (12)$$

Table 2 shows the measured and averaged values of $\mathcal{V}$ for the test signals, with different values of $M$. $N = 4096$ was used for both the calculation of $\mathcal{V}$ and for the time stretching's STFT. The values of $\mathcal{V}$ in Table 2 are given in decibels.

Table 2 shows that the vertical phase coherence is progressively lost as $M$ increases, which is the expected result. Interestingly, the changes are much larger between small values of $M$. However, on real musical signals, informal listening tests revealed a more regular progression. This can be explained by the pulse train-like nature of the test signals, and by the fact that $\mathcal{V}$ is an indirect approximation of vertical phase coherence.

| $M$ | 1 | 2 | 10 | 40 | 100 | 683 |
|---|---|---|---|---|---|---|
| $\mathcal{V}$ | -0.28 | -2.75 | -6.48 | -8.41 | -9.32 | -9.40 |

Table 2: Estimation of the vertical phase coherence $\mathcal{V}$ (in dB) of the presented algorithm for different values of $M$.

As a comparison, the value of $\mathcal{V}$ was measured as -9.7 for the unmodified phase-locked vocoder, and as -0.43 for time domain WSOLA. This closely matches the two measures of Table 2 corresponding to the two extreme values of $M$.

Listening tests confirmed that the presented algorithm is indeed perceptually similar to time domain techniques with $M = 1$, and to the phase-locked vocoder with $M = 683$. Listening tests also revealed that values of $M$ between 40 and 100 perceptually give the best results on signals with mixed content. Note that audio samples are available online [10].

Listening tests further revealed that on various real musical signals, vertical phase coherence was slightly better preserved with $M$ set to 2 or 3 rather than 1. A possible explanation is that the dominant peak of the band varies too much (frequently switching from main melody to bass line) when a single band is used to group the peaks that are locked together.

## 9 Further Improvements

While the presented algorithm allows one to arbitrarily choose the trade-off between horizontal and vertical phase coherence, it does not solve problems with transients that are common to both the phase vocoder and time domain techniques [1].

The phase-locked vocoder smears the transients, and time domain techniques drop or duplicate them. The presented technique just does something that is arbitrarily in between depending on $M$, which obviously remains objectionable.

Preliminary tests showed that some existing transient processing schemes [9], [14], [15] can be seamlessly incorporated in the presented algorithm to mitigate this problem.

## 10 Conclusions

This paper presented an audio time stretching technique that can arbitrarily control the trade-off between horizontal and vertical phase coherence using a single parameter. Formal measures on test signals as well as informal listening tests confirmed that the technique can sound anywhere between a phase-locked vocoder and a typical time domain technique. As such the proposed technique allows one to fine-tune the phase coherence for best results, depending on the audio signal being processed.

## References

[1] Jonathan Driedger and Meinard Müller, “A Review of Time-Scale Modification of Music Signals”, *Applied Science*, vol. 6, issue 2, pp. 57, 2016.

[2] Jean Laroche and Mark Dolson, “About This Phasiness Business”, *proc. of the International Computer Music Conference*, pp. 55-58, San Francisco, 1997.

[3] Jean Laroche and Mark Dolson, “Improved Phase Vocoder Time-Scale Modification of Audio”, *IEEE Transactions on Speech and Signal Processing*, vol. 7, no. 3, May 1999.

[4] Werner Verhelst, Marc 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*, April 1993.

[5] Sebastian Kraft, Martin Holters, Adrian von dem Knesebeck and Udo Zölzer, “Improved PVSOLA Time-Stretching and Pitch-Shifting for Polyphonic Audio”, *proc. of the 15th intl. conf. on Digital Audio Effects*, York, UK, September 2012.

[6] David Dorran, Eugene Coyle and Robert Lawlor, “Audio Time-Scale Modification using

a Hybrid Time-Frequency Domain Approach", *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, October 2005

[7] T. F. Quatieri and J. McAulay, "Shape invariant time-scale and pitch modification of speech", *IEEE Transactions on Signal Processing*, vol. 40, pp. 497–510, March 1992.

[8] Tan Roland K. C. and Lin Amerson H. J, "A Time-Scale Modification Algorithm Based on the Subband Time-Domain Technique for Broad-Band Signal Applications", *Journal of the Audio Engineering Society*, vol. 48, no. 5, pp. 437-449, May 2000.

[9] Jordi Bonada, "Automatic Technique in Frequency Domain for Near-Lossless Time-Scale Modification of Audio", *International Computer Music Conference*, pp. 396-399, 2000.

[10] Nicolas Juillerat, "Audio Time Stretching with Controllable Phase Coherence: Companion website", available online (as of March 2017) at www.pitchtech.ch/Confs/AES142

[11] Xavier Serra and Julius Smith III, "Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic Plus Stochastic Decomposition", *Computer Music Journal*, vol. 14, no. 4, pp. 12-24, Winter 1990.

[12] Jonathan Driedger, Meinard Müller, Sascha Dish, "Extending Harmonic-Percussive Separation of Audio Signals", *Proceedings of the 15th International Conference on Music Information Retrieval*, pp. 611-616, Taipei, Taiwan, 2014.

[13] Richard Füg, Andreas Niedermeier, Jonathan Driedger, "Harmonic-percussive-residual sound separation using the structure tensor on spectrograms", *IEEE Internal Conference on Acoustics, Speech and Signal Processing*, pp. 445-449, March 2016.

[14] Frederik Nagel and Andreas Walther, "A Novel Transient Handling Scheme for Time Stretching Algorithms", *127th Audio Engineering Society Convention*, New York, pp. 185-192, 2009.

[15] Nicolas Juillerat, Béat Hirsbrunner, "Audio Time Stretching with an Adaptive Multiresolution Phase Vocoder", *IEEE International Conference on Acoustics, Speech and Signal Processing*, New Orleans, LA, pp. 716-720, March 2017.