Rhythmic constant pitch time stretching for digital audio

Brendan TREVORROW[1];

[1] University of Southern Queensland, Australia

ABSTRACT

Constant pitch time stretching is not uncommon in audio editing software, however several issues arise when it is used on musical recordings, most notably the doubling and skipping of rhythmic transients. This paper examines three signal processing algorithms which are commonly used to provide constant pitch time stretching: these are SOLA (Synchronous Overlap and Add), TD-PSOLA (Time Domain Pitch Synchronous Overlap and Add), and Phase Vocoder. Enhancements to the SOLA and TD-PSOLA algorithms are provided which may make them more suited to rhythmic music. It is found that each of these three algorithms introduce audible artifacts in the time stretched waveform, the severity of these side effects and what causes them is also discussed.

Keywords: Digital Audio, Signal Processing, Music I-INCE Classification of Subjects Number(s): 74.9
(See `http://www.inceusa.org/links/Subj%20Class%20-%20Formatted.pdf`.)

1. TIME DOMAIN PITCH SYNCHRONOUS OVERLAP AND ADD

Time Domain Pitch Synchronous Overlap and Add (TD-PSOLA), first proposed in (1), is a time stretching technique which is commonly used in speech synthesis. This algorithm works by detecting the period T of a periodic waveform and splitting the waveform into overlapping frames of length $2T$, with each frame centred on an epoch (usually the peak) of the waveform and each frame multiplied by a Hanning window. To perform time expansion, a number of frames are duplicated to provide the required time ratio, conversely in time compression, frames are discarded. Figure 1 illustrates how the TD-PSOLA algorithm can be used to perform constant pitch time compression through the use of frame omission.

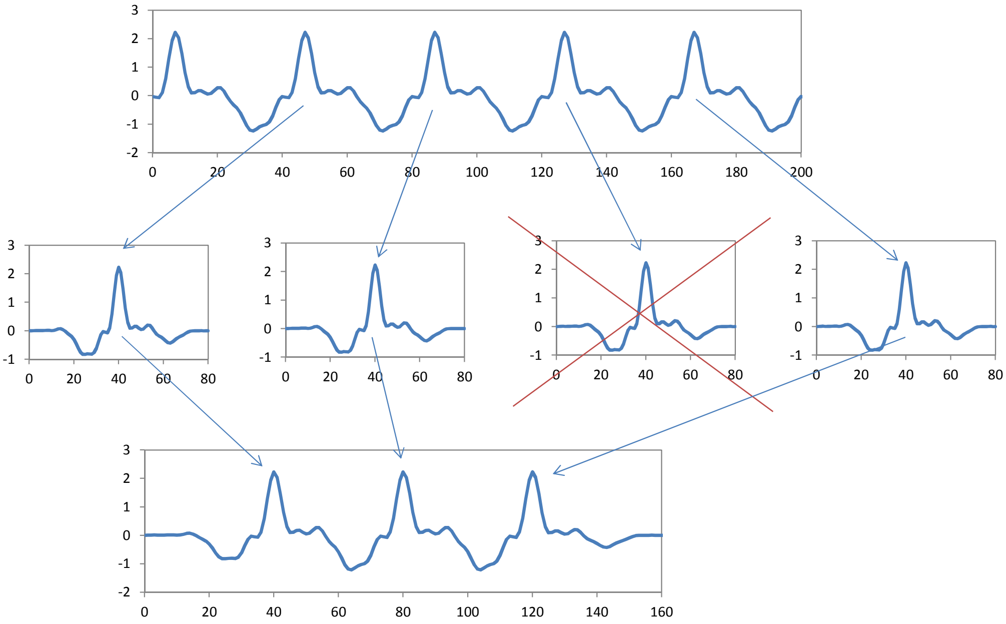

Figure 1 – TD-PSOLA algorithm for time compression.

[1] w0055187@umail.usq.edu.au

1.1 Modification of TD-PSOLA to Suit Rhythmic Time Stretching

The TD-PSOLA algorithm as described here cannot be applied to typical polyphonic musical recordings as its use applies to a periodic waveform (which a polyphonic recording is not), however the concepts of overlapping, duplicating and discarding frames can be used in a similar fashion. The approach that was investigated is to use frame lengths which divide evenly into the beat interval, and are at least as long as the period of the lowest audible frequency. This approach will be referred to from here on as "modified TD-PSOLA" although it should be mentioned that this is a bit of a misnomer as the algorithm is not pitch synchronous at all, rather it is based on tempo, and isn't synchronous.

To perform time expansion, every *n*th frame is required to be duplicated. Starting from the first frame and counting upwards, if a particular frame is reached, such that the next frame added gives the required ratio R, then that frame is duplicated. Mathematically this is:

$$\begin{aligned} R &= \frac{k+1}{k} \\ k &= \frac{1}{R-1} \end{aligned} \tag{1}$$

The value k calculated this way represents how often a duplicate frame is required, i.e. a duplicate frame is required every k frames, however it is very likely that this number will not be an integer. In order to ensure that the average ratio of the final stretched waveform is correct, it is necessary to maintain a variable which counts the number of frames which haven't been duplicated, when this variable exceeds k, then k is subtracted from it and the frame is duplicated.

For time compression, the value of k represents the number of frames between each frame skip, and is determined using similar reasoning:

$$k = \frac{1}{\frac{1}{R} - 1} - 1 \tag{2}$$

In equation (2), there is a -1 constant term at the end of the equation which is needed since it is assumed that the implementation includes the frame immediately after a deleted frame as being part of the frame skip (as it is in the implementation provided in (4)). If this is not the case then the -1 term should be removed, meaning k will be equal to the number of frames between deleted frames.

So far, the algorithm described does not make any allowances for any short transients which may exist only in a single frame (such a percussive hit). If a frame containing such a transient is duplicated or discarded, then it is possible that percussive hits in a musical recording will be doubled or skipped, which will be perceived as an abnormality to the listener. A simple allowance for the rhythm can be made if the tempo is known beforehand, by assuming that these transients will only occur at the start of every eighth, that is, the start of every beat and off-beat for a 4/4 time signature. If a frame is within a specified tolerance from one of these beat accents, then that frame is not eligible for duplication or disposal. Figure 2 illustrates this situation, where every 5th frame is discarded (time scale factor of 0.8 and $k = 3$).

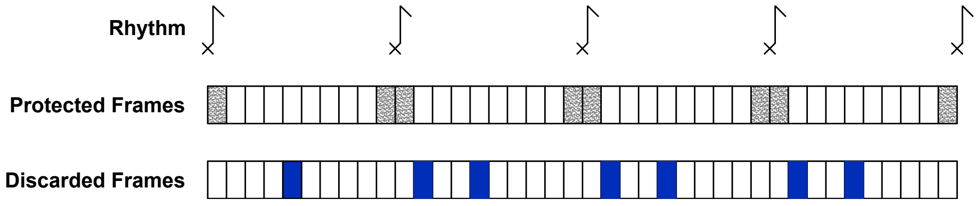

Figure 2 – Time stretching by 0.8 and allowing for rhythm.

2. SYNCHRONOUS OVERLAP AND ADD

The basis for the TD-PSOLA algorithm is the Synchronous Overlap and Add algorithm which was originally proposed in (2). The algorithm is an analysis/synthesis technique (similar to the phase vocoder) which splits the the source waveform into overlapping frames with centre spacing R_a and assembles an output waveform by overlapping these frames with a new centre spacing of $R_s = \alpha R_a$ where α is the required time scale ratio. Figure 3 shows diagrammatically how the algorithm can be used to achieve time expansion, with a factor of expansion, $\alpha = 1.333$ and factor of overlap, $\beta = 0.4$.

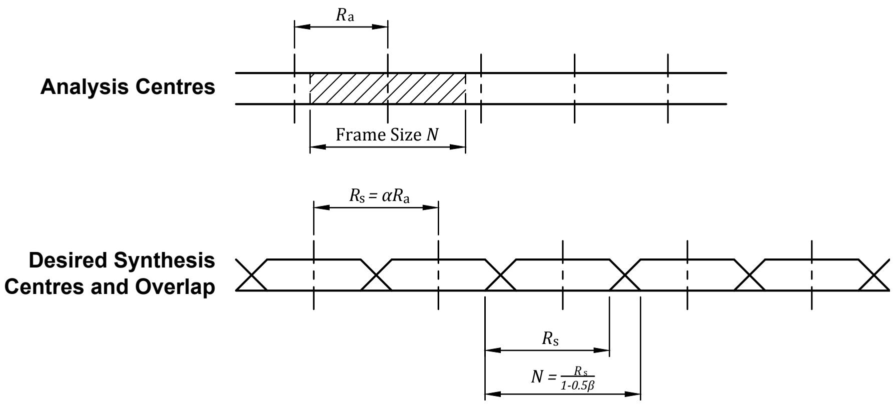
R_a
Analysis Centres
Frame Size N
$R_s = \alpha R_a$
Desired Synthesis Centres and Overlap
R_s
$N = \frac{R_s}{1\text{-}0.5\beta}$

Figure 3 – Time stretching using the SOLA algorithm.

To utilise this algorithm for time stretching of rhythmic audio, the analysis centre spacing R_a is chosen such that there are an integer number of frames between beat accents. The synthesis centre spacing is calculated using $R_s = \alpha R_a$ and from this, a frame size N is determined such that each synthesis frame overlaps by a specified amount.

To improve the phase coherence between overlapping frames in the output waveform, each frame in the synthesis stage is shifted by a small amount Δ_k such that the frames overlap at a point of maximum similarity. To determine the offset Δ_k, the cross correlation between the overlapping region in the previous frame y_{u-1} and the next frame y_u is used:

$$R_{y_u y_{u-1}}(\Delta_k) = \frac{1}{N} \sum_{n=0}^{N-1} y_u(n) y_{u-1}(n - \Delta_k) \tag{3}$$

The value of Δ_k which is maximum gives the point of maximum similarity, however because the previous frame only contains a finite number of samples before abruptly dropping to zero value, the maximum value of Δ_k that is used must reside in the left hand side of the cross correlation, to ensure that when the frames are overlapped, the envelope applied to the previous frame reaches zero before the end of the previous frame is reached, to prevent any sharp transitions from appearing in the output waveform.

3. PHASE VOCODER

The phase vocoder is an analysis/synthesis technique (see (3)), where the analysis stage essentially divides an input waveform into overlapping frames, with a standard spacing between frame centres. Each frame is multiplied by a windowing function and the Fourier Transform is calculated to give the Short Time Fourier Transform (STFT) representation. The synthesis stage involves modifying these STFT representations in the frequency domain, such that when the when the inverse Fourier transform is taken with the frames set a different centre spacing (to achieve a change in playback duration), the phases of the overlapping regions are properly aligned.

4. BEAT ALIGNMENT ANALYSIS

The three algorithms described were first tested for beat misalignment in the output waveforms, by time stretching from 140 beats per minute to 160 beats per minute (time compression factor of 0.875), which is a large change in tempo that would result in frequent beat skipping. The implementation of the TD-PSOLA and SOLA algorithms used can be found in (4) while the implementation of the phase vocoder was provided by (5).

The test procedure used is described in detail in (4), but to quickly summarise, the test waveform consisted of mostly silence except for short high frequency pulses with a Gaussian envelope, centred on every beat. To determine how well the time stretched waveform's beats aligned properly, the envelope of the time stretched waveform was multiplied with the envelope of the desired waveform. The results of this test for the modified PSOLA and SOLA algorithms is shown in Figures 4 & 5.

The vertical axes of these graphs represent a measure of how well the beat pulses have aligned, that is, a beat with a peak value of unity can be considered to have perfectly aligned with where it should be, while an

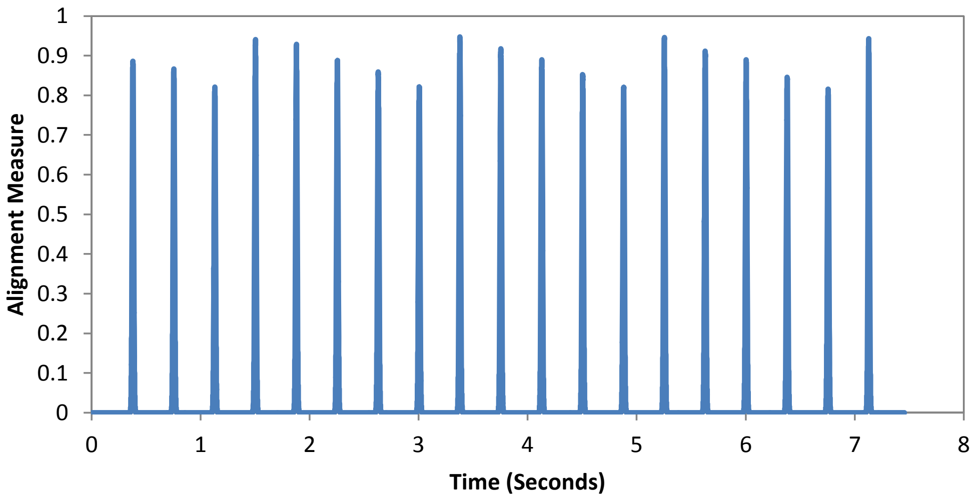

Figure 4 – Beat alignment of modified PSOLA algorithm.

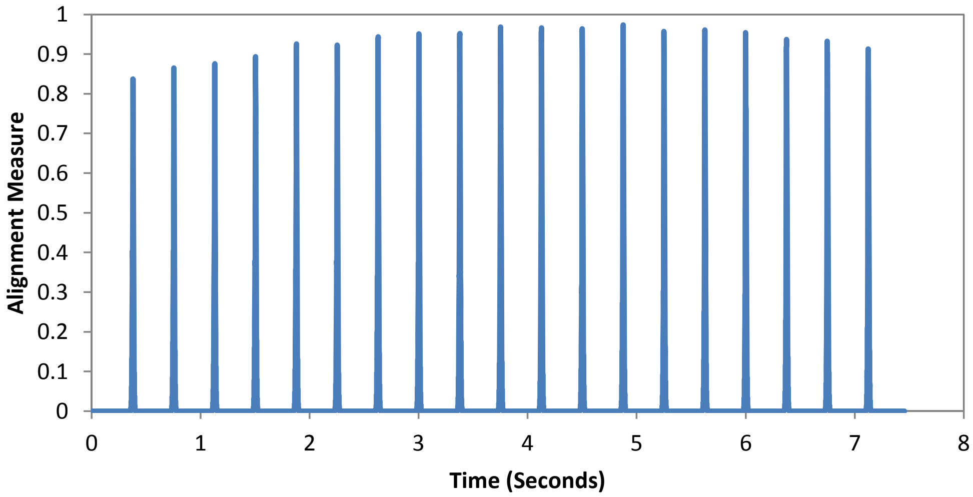

Figure 5 – Beat alignment of SOLA algorithm.

absence of a peak would mean complete misalignment. Looking at Figure 4, there appears to be a repeating pattern in the alignment, with the amount of misalignment increasing for a few beats before resetting. This is a side effect of using frame omission to achieve time stretching, which implies that up to one frame width of misalignment can be expected as frames are omitted. No such pattern is observed in Figure 5, which aligns each frame precisely where they should appear, minus a small time delay for positioning at a point of maximum similarity, which would account for the slight misalignments occurring toward the beginning of the waveform.

The beat alignment of the two overlap-add algorithms shown so far have performed reasonably well. In contrast to this, the beat alignment of the phase vocoder implementation is shown in Figure 6.

The absence of peaks in the 1 to 2 seconds region and 3 to 5 seconds region suggests that this algorithm has failed to align the beats in these positions correctly. To confirm whether this is actually true, a time domain plot of the output of the phase vocoder is shown in Figure 7, where the blue line represents the phase vocoder time stretched waveform and the light purple line represents what the time stretched waveform should actually be.

Examining Figure 7 reveals that there is an amplitude modulation occurring as well as beat doubling. This could be explained by the fact that this particular phase vocoder implementation works on frame sizes which are powers of 2 (in this situation 1024 was used), which do not divide into the number of samples per beat evenly, resulting in beat accents which are sometimes split across frames.

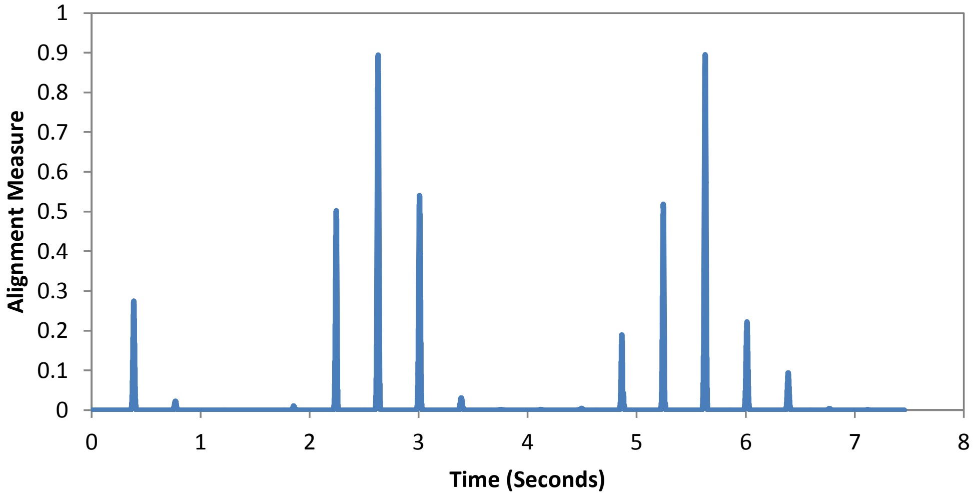

Figure 6 – Beat alignment of phase vocoder algorithm.

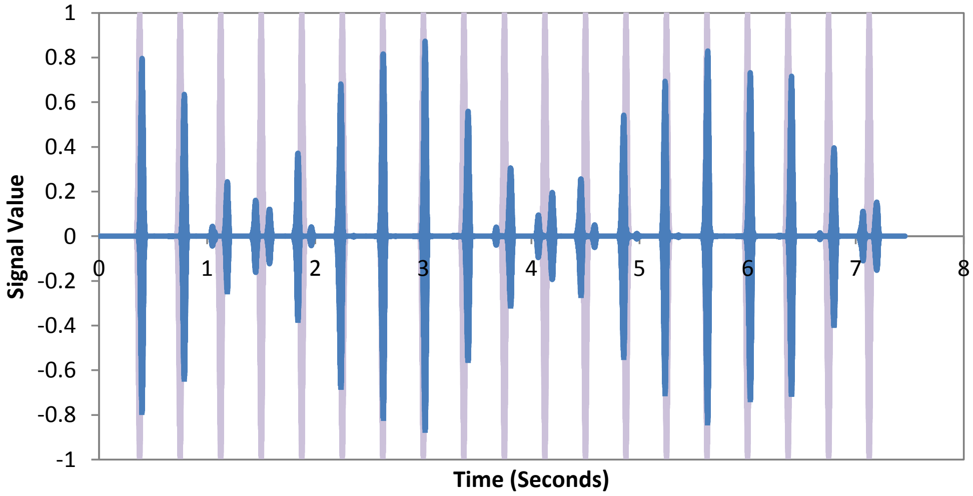

Figure 7 – Output of phase vocoder time compression.

5. SUBJECTIVE ANALYSIS

In addition to the quantitative analysis provided by the beat alignment test, a subjective test was performed, where each algorithm was used to time stretch electronic dance tracks (see Appendix A for track list). These tracks were chosen as they contain strong and distinctive percussive rhythms (which would reveal issues with short transients and beat alignment), contain notes of differing lengths across most of the frequency spectrum (to test the performance of frame overlap) and contain no audible distortion which could be attributed to other sources (such as clipping). The structure of these tracks was also advantageous, as they contain sparse intros and outros consisting mostly of percussive rhythm, and a mid section which lacks the percussive rhythm, but contains mostly melodic and harmonising parts. This meant that issues with the percussive rhythm (beat alignment) could effectively be isolated from issues with frame overlap. Out of the four tracks listed above, only the last track contains vocals while the rest are purely instrumental.

The test involved both time compression and expansion of all tracks, where a time ratio resulting in a tempo change of 8 beats per minute was used. The most prominent artifacts encountered in each of the algorithms, as well as their severity, are summarised in Table 1.

With the phase vocoder, the most prominent artifact was distortion of the percussive rhythm. This would be caused by the fact that the underlying model behind the phase vocoder assumes that the waveform can be modelled by discrete sinusoids which are coherent across multiple frames, which is not true in the case of these percussive instruments, whose duration is comparable to the frame length. The result is that these short transients appear to be “smeared” across multiple frames. This effect can be reduced by using shorter STFT

Table 1 – Summary of Audible Time Stretching Artifacts

Algorithm	Artifact	Severity
phase vocoder	transient smearing	moderate at 1024 samples
modified PSOLA	amplitude modulation	slight
SOLA	percussive flanger effect	moderate

frames, however this reduces the frequency resolution of the algorithm.

The most prominent artifact in the proposed modifications to the TD-PSOLA algorithm is that of modulation of background harmonising instruments as well as slight modulation of the vocals. This occurs in the regions where a frame is neighbouring a duplicated or omitted frame and is caused by destructive interference as a result of the phases of the overlapping waveforms not being properly aligned. The severity of this type of artifact is only slight, but once noticed is difficult to ignore.

The standard SOLA algorithm does not contain the above mentioned modulation artifact, however this algorithm tends to distort the percussive instruments, which end up sounding as if they have had a “flanger” effect applied to them. The severity of this artifact can be controlled by varying the parameters of the algorithm, with more overlap and smaller frame sizes producing less of this distortion.

6. FURTHER WORK

Constant pitch time stretching algorithms continue to be the subject of research: while this project was mostly interested in those suitable for music recordings, it was necessary to limit its scope. There are a number of other approaches which could have been investigated, of these, perhaps the most promising is an analysis/synthesis technique based on the wavelet transform. The wavelet transform does not suffer from the time/frequency resolution trade-off encountered with the phase vocoder’s short time Fourier transform and thus an algorithm based on it may yield better results.

The algorithms investigated were reliant on the start of the waveform being aligned with the beginning of a beat and made no allowance for changes in tempo within the recording. This restriction could be removed by combining the time stretching algorithm with a beat detection algorithm, although again, these have their own limitations, which could perhaps be investigated as an extension to the current work.

7. CONCLUSIONS

Three algorithms for constant pitch time stretching have been examined. These were Time Domain Pitch Synchronous Overlap and Add (with modification to suit tempo), Synchronous Overlap and Add and the Phase Vocoder.

The phase vocoder does not perform well when time stretching audio which contains short transients due to the inherent assumption that each frame contains sinusoids which are coherent across multiple frames. It also has issues with beat alignment, which suggests that further work is required perfecting the phase unwrapping part of the phase vocoder. On the other hand, the two overlap-add methods were capable of aligning the beat accents within the specified tolerance, however the two overlap-add methods proposed here each contain their own unique audible artifacts, the severity of which can be controlled by varying parameters.

ACKNOWLEDGEMENTS

I would like to thank my final year project supervisor, John Leis, whose continuous support and insight helped lead me to approach my project in ways I would not have considered by myself. I would also like to thank Michael Hayne and Ian Hillock of the Australian Acoustics Society Queensland Division for their recognition of my final year project as well as providing me with the opportunity to present at the April 2014 Queensland Division technical meeting.

REFERENCES

1. Hamon C, Mouline E, Charpentier F. A Diphone Synthesis System Based on Time Domain Prosodic Modifications of Speech. In: Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Glasgow, Scotland; 1989. p. 238–241.

2. Roucos S, Wilgus A. High quality time-scale modification for speech. In: Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP ’85.. vol. 10; 1985. p. 493–496.

3. Laroche J, Dolson M. Improved Phase Vocoder Time-Scale Modificataion of Audio. IEEE Transactions on Audio and Speech Processing. 1999;7(3):323–332.

4. Trevorrow B. Investigation of Digital Audio Manipulation Methods. Toowoomba: University of Southern Queensland; 2013. `http://eprints.usq.edu.au/id/eprint/24622`.

5. Ellis DPW. A Phase Vocoder in Matlab; 2002. [Online; accessed August-2013]. `http://www.ee.columbia.edu/ln/rosa/matlab/pvoc/`.

A. SUBJECTIVE TEST AUDIO SAMPLES

The following tracks were used in the subjective test detailed in section 5.

Table 2 – Tracks used in subjective test

Mat Zo - The Fractal Universe
Timo Pralle - Hopeful (Kim Svärd Remix)
Ronny K. vs Ziki - Memories (New World Remix)
John O’Callaghan feat. Josie - Out of Nowhere