2003-01-01

An Efficient Audio Time-scale Modification Algorithm for use in a Subband Implementation

David Dorran
Technological University Dublin, david.dorran@tudublin.ie

Robert Lawlor
National University of Ireland, Maynooth

Recommended Citation

Dorran, D. & Lawlor, R. (2003) An efficient audio time-scale modification algorithm for use in a subband implementation. *International Conference on Digital Audio Effects, London, England, 2003,*

AN EFFICIENT AUDIO TIME-SCALE MODIFICATION ALGORITHM FOR USE IN A SUBBAND IMPLEMENTATION

David Dorran

Dept. of Electronic Engineering
Dublin Institute of Technology
david.dorran@dit.ie

Robert Lawlor

Dept. of Electronic Engineering
National University of Ireland, Maynooth.
rlawlor@eeng.may.ie

ABSTRACT

The PAOLA algorithm is an efficient algorithm for the time-scale modification of speech. It uses a simple peak alignment technique to synchronise synthesis frames and takes waveform properties and the desired time-scale factor into account to determine optimum algorithm parameters. However, PAOLA has difficulties with certain waveform types and can result in poor synchronisation for subband implementations. SOLA is a less efficient algorithm but resolves the issues associated with PAOLA's implementation. We present an algorithm that is a combination of the two approaches that proves to be an efficient and effective algorithm for a subband implementation.

1. INTRODUCTION

Time-scale modification of audio alters the duration of an audio signal while retaining the signals local frequency content, resulting in the overall effect of speeding up or slowing down the perceived playback rate of a recorded audio signal without affecting the quality, pitch or naturalness of the original signal. This facility is useful for such applications as enhancement of degraded speech, language and music learning, fast playback for telephone answering machines and altering the tempo of recorded music so as to integrate synchronously with scenes within the film industry.

Altering the time-scale of an audio signal can be achieved in the time-domain or frequency-domain with advantages and disadvantages associated with each approach. Frequency-domain techniques generally fall into one of two categories, phase vocoder [1] and sinusoidal modeling [2], and are capable of applying high quality time-scale modifications to a variety of complex audio signals within a wide range of time-scale factors, but their versatility comes at the expense of their computational requirements. Computationally efficient time-domain techniques operate by simply discarding or repeating suitable segments of the audio signal. The discard/repeat process relies heavily upon the existence of a quasi-periodic waveform, making time-domain approaches suitable for speech and monophonic music but unsuitable for most polyphonic music due to the generally complex multi-pitch nature of the waveform. However, the subband analysis synchronised overlap-add (SASOLA) [3] and subband waveform similarity overlap-add (subband WSOLA) [4] algorithms have demonstrated that applying time-domain time-scale modification algorithms on a subband basis can resolve this issue.

In this paper we discuss the matters arising from a subband implementation and describe an efficient time-scale modification algorithm that is suitable for use within a subband implementation. Section 2 summarises the commercially popular synchronised overlap-add (SOLA) [5] and the efficient peak alignment overlap-add (PAOLA) [6] algorithms and also outlines a variant of SOLA, the synchronised and adaptive overlap-add (SAOLA) [7] algorithm, which improves upon the quality of SOLA for high time-scale factors and provides a reduction in computational requirements for low time-scale factors. In section 3 we briefly describe the operation of the SASOLA and subband WSOLA approaches. Section 4 highlights the advantages and limitations of both SOLA and PAOLA; and introduces the variable-parameter synchronised overlap-add (VSOLA) algorithm, which takes advantage of the best features of the SOLA and PAOLA algorithms to form a computationally efficient algorithm suitable for a subband implementation. Sections 5 and 6 present a comparison between VSOLA and SAOLA in terms of computational requirements and output quality, respectively. Section 7 concludes the paper.

2. SOLA, SAOLA AND PAOLA

2.1. SOLA

The SOLA algorithm segments the input signal x into overlapping frames, of length N samples, the start of the m^{th} input frame being positioned at mS_a samples along the input. S_a is the analysis step size. The time-scaled output y is synthesised by overlapping successive frames with the start of the m^{th} frame positioned at $mS_s + k_m$ samples along the output. S_s is the synthesis step size, and is related to S_a by $S_s = \alpha S_a$, where α is the time-scaling factor. k_m is a deviation allowance that ensures that successive synthesis frames overlap in a synchronous manner. k_m is chosen such that

$$R_m(k) = \frac{\sum_{j=0}^{L_m-1} y(mS_s + k + j)x(mS_a + j)}{\sqrt{\sum_{j=0}^{L_m-1} x^2(mS_a + j)\sum_{j=0}^{L_m-1} y^2(mS_s + k + j)}} \qquad (1)$$

is a maximum for $k = k_m$, where m represents the m^{th} input frame and L_m is the length of the overlapping region. k is in the range $k_{min} \leq \ k \leq k_{max}$. Typically, N is fixed at 30ms for speech and 40ms for music, S_a is in the range of $N/3$ to $N/2$, k_{min} is $-N/2$ and k_{max} is $N/2$.

$R_m(k)$ is a correlation function which ensures that successive synthesis frames overlap at the 'best' location i.e. that location where the overlapping frames are most similar. Having located the 'best' position at which to overlap, the overlapping regions of the frames are weighted prior to combination, generally using a linear or raised-cosine function. The output is then given by

$$y(mS_s + k_m + j) :=$$

$$(1 - \mathrm{f}(j))y(mS_s + k_m + j) + \mathrm{f}(j)x(mS_a + j), 0 \le j \le L_m - 1 \quad \text{(2a)}$$

$$y(mS_s + k_m + j) = x(mS_a + j), L_m \le j \le N - 1 \quad \text{(2b)}$$

where := in equation (2a) means 'becomes equal to' and $f(j)$ is a weighting function such that $0 \le f(j) \le 1$.

A linear weighting function can be expressed as

$$\mathrm{f}(j) = 0, j < 0 \quad \text{(3a)}$$

$$\mathrm{f}(j) = j / (L_m - 1),\ 0 \le j \le L_m - 1 \quad \text{(3b)}$$

$$\mathrm{f}(j) = 1, j > L_m - 1 \quad \text{(3c)}$$

2.2. SAOLA

In general the parameters N, S_a, k_{min} and k_{max} are fixed for SOLA at algorithm development, which can be problematic. Consider the case where S_a is fixed at $N/3$, k is in the range 0 to $N/2$ and k_m for the previous iteration was 0. If $\alpha = 2$ then $S_s = 2N/3$. For this case the number of possible overlaps is limited to $N/3$ i.e. from an overlap of $N/3$ to an overlap of 1. By limiting the number of possible overlaps the output quality is degraded. It can easily be shown that the number of possible overlaps is less than $N/2$ for $\alpha > 1.5$. This problem could be alleviated by allowing k be in the range $-N/2$ *to* $N/2$. For this case, the number of possible overlaps is less than $N/2$ for $\alpha > 3$. However, the number of possible overlaps is greater than $N/2$ for $\alpha < 3$ and equal to N for $\alpha \le 1.5$. In [7] it is shown that $N/2$ possible overlaps provides an adequate search range and any number greater than this increases the computational load unnecessarily. From above, S_s should ideally be $N/2$ for all α, allowing $N/2$ possible overlaps for all α, when k is in the range of $N/2$ to 0. SAOLA achieves this by allowing S_a be adaptive i.e.

$$S_a = N/(2\alpha) \quad \text{(4)}$$

This result also has the effect of reducing the number of computations required for low time-scale factors.

2.3. PAOLA

PAOLA also segments the input waveform into overlapping analysis/input frames of length N separated by a distance S_a. During synthesis the first input frame is copied to the output, to become the current output. For subsequent input frames, the maximum peaks are located in the last SR samples of the current output and the first SR samples of the current input frame, where SR is the search region and corresponds to one cycle of the lowest likely fundamental component of the input signal. Peaks are then aligned so that frames overlap synchronously. The overlapping regions of the frames are weighted prior to combination using a linear function.

PAOLA determines optimum analysis parameters by considering two extreme situations. The first case considers the situation where a peak is found in the last element of the current output and first element of the current input frame, as illustrated in figure 1 (c). For this case the analysis-overlapping region is almost repeated, except for one sample. For high quality time-scale modification the repeated segment should be short enough to ensure quaisi-stationarity during voiced regions, so

$$N - S_a \le L_{stat} \quad \text{(5)}$$

where L_{stat} is that length that ensures that the segment is quaisi-stationary during voiced regions. Since $N = SR + S_s$ and $S_s = \alpha S_a$

$$(\alpha - 1)S_a \le L_{stat} - SR \quad \text{(6)}$$

So,

$$S_a \le \frac{L_{stat} - SR}{\alpha - 1} \quad \text{for } \alpha > 1 \quad \text{(7a)}$$

and

$$S_a \ge \frac{L_{stat} - SR}{\alpha - 1} \quad \text{for } \alpha < 1 \quad \text{(7b)}$$

Now consider the case where a peak is located in the first element of the search region SR of the current output and the last element of the search region of the current input frame i.e. maximum overlap. This case is illustrated in figure 1 (d). For this case a segment of length S_a – *(*S_s – *SR)* is discarded during synthesis. For high quality time-scale modification the discarded segment should be short enough to ensure quaisi-stationarity during voiced regions so

$$S_a - (S_s - SR) \le L_{stat} \quad \text{(8)}$$

Since $S_s = \alpha S_a$

$$(1 - \alpha)S_a \le L_{stat} - SR \quad \text{(9)}$$

So,

$$S_a \ge \frac{L_{stat} - SR}{1 - \alpha} \quad \text{for } \alpha > 1 \quad \text{(10a)}$$

and

$$S_a \le \frac{L_{stat} - SR}{1 - \alpha} \quad \text{for } \alpha < 1 \quad \text{(10b)}$$

Combining (7a) and (10a) gives

$$\frac{L_{stat} - SR}{\alpha - 1} \ge S_a \ge \frac{L_{stat} - SR}{1 - \alpha} \quad \text{for } \alpha > 1 \quad \text{(11a)}$$

Combining (7b) and (10b) gives

$$\frac{L_{stat} - SR}{1 - \alpha} \ge S_a \ge \frac{L_{stat} - SR}{\alpha - 1} \quad \text{for } \alpha < 1 \quad \text{(11b)}$$

The number of iterations that are executed is inversely proportional to S_a, therefore S_a should be maximised giving

$$S_a = \frac{L_{stat} - SR}{|1 - \alpha|} \quad \text{for all } \alpha \quad \text{(12)}$$

And since $N = SR + \alpha S_a$

$$N = SR + \alpha\left(\frac{L_{stat} - SR}{|1 - \alpha|}\right) \quad \text{for all } \alpha \quad \text{(13)}$$

Equations (12) and (13) provide optimum analysis parameters for PAOLA's implementation.

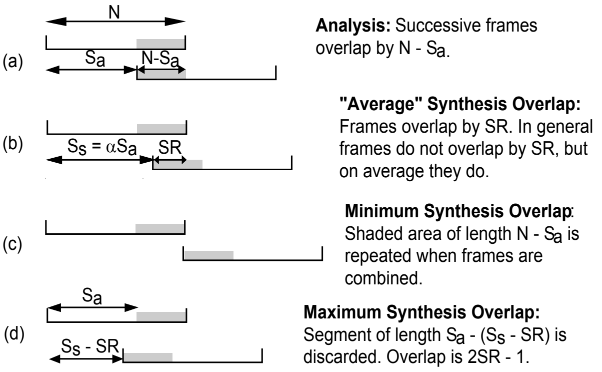
N
Sa
N-Sa
(a)
Analysis: Successive frames overlap by N - Sa.
(b)
Ss = αSa
SR
"Average" Synthesis Overlap: Frames overlap by SR. In general frames do not overlap by SR, but on average they do.
(c)
Minimum Synthesis Overlap: Shaded area of length N - Sa is repeated when frames are combined.
(d)
Sa
Ss - SR
Maximum Synthesis Overlap: Segment of length Sa - (Ss - SR) is discarded. Overlap is 2SR - 1.

Figure 1. *PAOLA analysis and synthesis.*

3. SUBBAND APPROACH IMPLEMENTATION AND ISSUES

As mentioned in the introduction time-domain time-scale modification techniques rely upon the existence of a strong quasi-periodic element within the signal to be time-scaled in order to achieve high quality results. Certain types of signal, such as polyphonic music, may not contain a strong quasi-periodic element and are therefore unsuitable for time-scale modification directly in the time-domain, however applying time-domain techniques on a subband basis can resolve this issue. The major issues concerning a subband approach are the partitioning of a complex waveform into subbands of lesser complexity, that are suitable for time-scale modification in the time-domain, and the recombination of the time-scaled subbands in a synchronous manner. The solutions to these issues are diametrically opposite since partitioning a complex waveform into many subbands reduces the complexity of each subband but increases potential subband synchronisation problems and vice versa.

Subband synchronisation problems occur because time-domain time-scale modification techniques require a deviation allowance to ensure that successive synthesis frames overlap in a synchronous manner. Each subband will almost certainly require different deviation allowances, resulting in poorly synchronised subbands. The subband synchronisation problem can be simulated by first partitioning the signal into subbands; then passing each subband through a random delay ranging from 0 to some maximum delay, d_{max}. By considering a trivial case where d_{max} is set to 1 hour the synchronisation problem is highlighted, since delay differences between subbands of up to one hour would certainly introduce audible artifacts. The delays mentioned in our simulation model correspond to deviation allowances within time-domain algorithms, therefore subband synchronisation problems can be reduced by decreasing the search regions of the time-domain algorithms, however decreasing the search region can have a negative affect on the quality of each time-scaled subband since there is a minimum search range required in order to identify a suitable overlap position. In [8] these types of group/subband delays are discussed in more detail.

Both SASOLA and subband WSOLA operate by first filtering the complex input waveform into subbands before applying a time-domain time-scale modification algorithm to each subband. The resulting time-scaled subbands are then summed, producing a high quality time-scaled version of the original multi-pitched signal, as illustrated in figure 2. SASOLA partitions broadband audio signals sampled at 44.1 kHz into subbands using a 17-channel cosine-modulated, perfect reconstruction, uniform width filterbank. The SOLA algorithm is then applied to each subband using a 40ms frame on all subbands for time-scale compression; for time-scale expansion a 40ms frame is used on the lowest frequency subband and a 20ms frame on all other subbands. Subband WSOLA partitions audio signals sampled at 10kHz into subbands using a 16-channel, perfect reconstruction, uniform width filterbank. The waveform similarity overlap-add [9] (WSOLA) algorithm is then applied to each subband using smaller frame lengths for higher frequency subbands (values not provided).

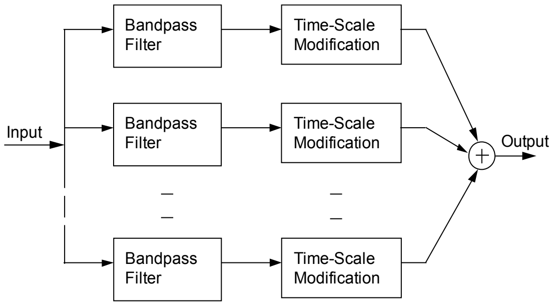

Figure 2. *Subband approach to time-scale modification.*

4. VSOLA

Although more efficient than SOLA, the PAOLA algorithm has difficulties with certain waveform types and subband implementations. Consider the situation shown in figure 3(a), which illustrates two overlapping segments of a speech waveform. The PAOLA algorithm operates by aligning the peaks of the current output and the current synthesis frame before summing, with the use of a linear cross-fade function, resulting in a high quality output as shown in the lower waveform of figure 3(a). Now consider the situation shown in figure 3(b), which illustrates two overlapping segments of a trombone waveform. Once again the PAOLA algorithm aligns the peaks of the current output and current synthesis frame. However, for this case the peak alignment procedure fails to overlap at the correct position, resulting in a poor quality output. If a SOLA type correlation function were used in the alignment process this issue, which we dub the peak ambiguity problem, would not arise.

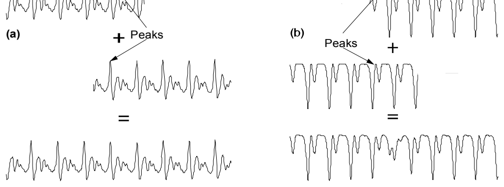

Figure 3: *PAOLA peak ambiguity problem.*

The PAOLA algorithm also poses a potential problem for a subband implementation since it relies upon an averaging effect to ensure that the final output is α times the length of the input signal and cannot provide a guarantee as to the length of the current output after a given number of iterations. For a PAOLA implementation the delay differences between subbands can potentially range from 0 to $2mSR$, where m represents the m^{th} iteration of the algorithm. Relying on an average overlap in this way is suitable for most signals but introduces noticeable synchronisation problems at a subband level. Inadequate synchronisation of subbands is particularly noticeable at transients and results in transients sounding metallic. A SOLA based approach, however, provides a guarantee that the length of the current output after m iterations is within the range $m*S_s + N + k_{min}$ to $m*S_s + N + k_{max}$. This level of control of the output length and, therefore, the inter-subband delay differences is crucial for the successful implementation of a subband approach.

Although equations (12) and (13) were derived for the PAOLA algorithm, it can be shown that the principles on which the derivation of these parameters was based also apply to SOLA if we consider that the overlap between the m^{th} and $(m\text{-}1)^{th}$ synthesis frames, as illustrated by figure 4, is given by:

$$OL = N - S_s + k_{m-1} - k_m \tag{14}$$

If we define the search region SR to be $k_{max} - k_{min}$, the maximum overlap is then $N - S_s + SR$, i.e. when $k_{m-1} = k_{max}$ and $k_m = k_{min}$, which is the situation illustrated in figure 1(d). The minimum overlap is $N - S_s - SR$ i.e. when $k_{m-1} = k_{min}$ and $k_m = k_{max}$, which, since $N = S_s + SR$, is illustrated in figure 1(c). Equations (12) and (13) can then be derived for SOLA in the same way as they were for PAOLA in section 2.

It should be noted however that the search range SR should be twice that of PAOLA for SOLA, so that an suitable overlap position can be identified using correlation, as can be understood from [7], allowing (12) and (13) be used in determining the corresponding parameters for SOLA's implementation. It should also be noted that the length of L_{stat} can be relaxed for a SOLA based implementation since the correlation function used helps ensure that only segments of suitable length will be discarded/repeated. In PAOLA's implementation this is not the case since only maximum peaks are used to identify the length of segment to be discarded/repeated and so a suitably small value of L_{stat} must be used. For the purpose of discrimination we will call the variant of SOLA that uses equations (12) and (13) to determine the window length and analysis step size VSOLA (variable-parameter synchronised overlap-add). Since VSOLA operates in the same way as SOLA (once S_a and N are determined) it can also take advantage of the computational savings set out in [10] and [11]. In our implementation we set $k_{min} = 0$, therefore $k_{max} = SR$. For a non-subband implementation SR is set to 15ms and 20ms for speech and monophonic music, respectively. To minimize potential subband synchronisation problems for a subband implementation we used smaller values for SR for higher frequency subbands. Using the same cutoff frequencies as SASOLA we set SR equal to 5ms, 10ms, 15ms and 20ms for subbands with lower cutoff frequencies greater than 15kHz, 10kHz, 5kHz and 0Hz, respectively. For all cases we found that setting $L_{stat} = 5SR/3$ produced high quality results.

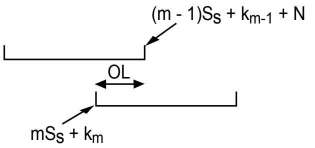

Figure 4: *Overlap between successive SOLA synthesis frames.*

5. VSOLA/SAOLA COMPUTATIONAL LOAD COMPARISON

Equations (12) and (13) provide optimum analysis parameters for SOLA's implementation and simply results in a reduction in the total number of iterations required for the algorithms implementation. Since the total number of iterations, I, required for signal of length L_x is given by

$$I = L_x/S_a \tag{15}$$

The ratio of SAOLA to VSOLA computational operations can then be shown to be

$$\frac{I_{SAOLA}}{I_{VSOLA}} = \frac{2(L_{stat} - SR)}{N} \times \frac{\alpha}{|1 - \alpha|} \tag{16}$$

Figure 5 illustrates the ratio of SAOLA to VSOLA operations for time-scale factors ranging from 0.5 to 3, with N = 30ms, SR = 15ms and L_{stat} = 25ms.

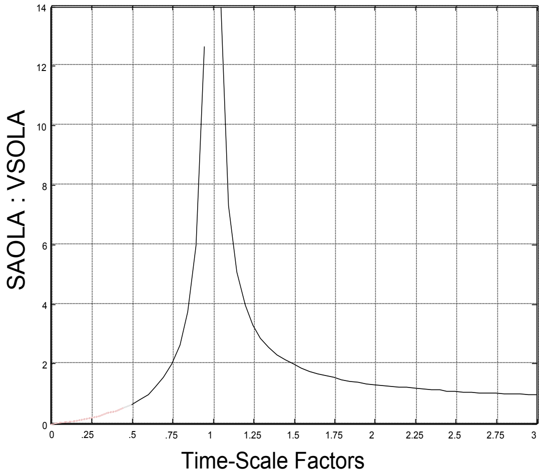

Figure 5: *Ratio SAOLA to VSOLA Computations.*

6. VSOLA/SAOLA OUTPUT QUALITY COMPARISON

10 evaluation subjects of various age and gender carried out informal listening tests. The test comprised of 10 comparisons between a track time-scaled by SAOLA and the same track time-scaled by VSOLA, using the same time-scale factor. The subjects were not informed which track was a SAOLA time-scaled track or which was a VSOLA time-scaled track. The tests covered a selection of time-scale factors ranging from 0.5 to 3 and comprised of speech and both monophonic and polyphonic music signals. The polyphonic music signals were time-scaled

using a subband approach using the same filterbank cutoff frequencies as SASOLA's implementation. The parameters used for VSOLA's implementation were the same as those set out in section 4 and for SAOLA the *N* parameter was set to twice VSOLA's *SR* parameter.

The listening test results, summarised in table 1, show that the output quality of signals time-scaled by SAOLA and VSOLA are approximately equal.

Subjects Indication	**%**
SAOLA much better than VSOLA	0 %
SAOLA slightly better than VSOLA	20 %
SAOLA equal to VSOLA	47 %
SAOLA slightly worse than VSOLA	32 %
SAOLA much worse than VSOLA	1 %

Table 1. *Summary of listening test results.*

7. CONCLUSION

PAOLA is an efficient algorithm for the time-scale modification of speech but is unsuitable for a subband implementation due to subband synchronisation and peak ambiguity issues. SOLA is less efficient than PAOLA, however it has proved to be a suitable algorithm for a subband implementation. This paper presents an algorithm, VSOLA, which takes advantage of the best features of the SOLA and PAOLA to produce an efficient algorithm suitable for use within a subband implementation. Listening tests have shown that VSOLA and an adaptive version of SOLA, SAOLA, produce a time-scaled output of the same quality for both subband and non-subband implementations.

8. REFERENCES

[1] Laroche, J., Dolson, M., "Improved phase vocoder time-scale modification of audio*", IEEE Transactions on Speech and Audio Processing*, vol. 7, issue 3, pp. 323 -332, May 1999.

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

[3] Tan, R.K.C. and Lin, A.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.

[4] Spleesters, G. and Verhelst, W. and Wahl, A., "On the application of automatic waveform editing for time warping digital and analog recordings", *Proc. 96th Audio Engineering Society Convention*, Amsterdam, preprint 3843, 1994.

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

[6] Dorran D., Lawlor, R. and Coyle E., "High Quality Time-Scale Modification of Speech using a Peak Alignment Overlap-Add Algorithm (PAOLA)", *IEEE International Conference on Acoustics, Speech and Signal Processing*, Hong Kong, paper no. 2382, April 2003.

[7] Dorran D., Lawlor, R. and Coyle E., "Time-Scale Modification of Speech using a Synchronised and Adaptive Overlap-Add (SAOLA) Algorithm", *Audio Engineering Society 114th Convention 2003,* Amsterdam, The Netherlands, preprint no. 5834, March 2003.

[8] J. Blauert and P. Laws, "Group Delay Distortions in Electroacoustical Systems", *Journal of the Acoustical Society of America*, vol. 63, no. 5, pp. 1478-1483, May 1978.

[9] Verhelst, W., Roelands, M., **"**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. 554 –557, 1993.

[10] Wong, P.H.W., Au, O.C., "Fast SOLA-based time scale modification using modified envelope matching", *IEEE International Conference on Acoustics, Speech and Signal Processing,* vol. 3*, pp.* 3188- 3191, 2002.

[11] Yim, S., Pawate, B.I., "Computationally efficient algorithm for time scale modification (GLS-TSM)", *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 1009 –1012, 1996.