2003-01-01

Time-Scale Modification of Speech Using a Synchronised and Adaptive Overlap-add (SAOLA) Algorithm

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

Robert Lawlor
National University of Ireland, Maynooth

Eugene Coyle
Technological University Dublin, Eugene.Coyle@tudublin.ie

Recommended Citation

Dorran, D., Lawlor, R. & Coyle, E. (2003) Time-scale modification of speech using a synchronised and adaptive overlap-add (SAOLA) algorithm. *114th. Audio Engineering Society Convention, paper#5834, Amsterdam, the Netherlands, 2003.*

Audio Engineering Society

Convention Paper

Presented at the 114th Convention
2003 March 22–25 Amsterdam, The Netherlands

Time-Scale Modification of Speech using a Synchronised and Adaptive Overlap-Add (SAOLA) Algorithm

David Dorran[1], Bob Lawlor[2] and Eugene Coyle[1]

[1]Dublin Institute of Technology, Kevin Street Lower, Dublin 8, Ireland.

[2]National University of Ireland, Maynooth, County Kildare, Ireland.

ABSTRACT

The synchronised overlap-add (SOLA) algorithm is a commercially popular and considerably researched audio time-scale modification technique. It operates in the time domain and uses a correlation technique to ensure that synthesis frames overlap in a synchronous manner. We present a modification to SOLA that allows the analysis step size adapt to the desired time-scale factor. The synchronised and adaptive overlap-add (SAOLA) algorithm improves upon the output quality of SOLA for high time-scale factors and reduces the computational requirements for low time-scale factors. However, the computational requirements for high time-scale factors are increased.

1. INTRODUCTION

Time-scale modification of speech allows the rate of articulation of a speech passage be increased or decreased, ideally without affecting the quality, pitch or naturalness of the original signal. This facility is useful for such applications as foreign language learning and fast playback for telephone answering machines.

The synchronised overlap-add (SOLA) algorithm [1] is a commercially popular time domain technique, which we summarise in section 2. In section 3 we develop a refinement to SOLA which allows the analysis step size adapt to the desired time-scale factor, resulting in a considerable improvement in the quality of output for large time-scale factors, at the expense of an increase in the computational requirements, and a reduction in the number of computations required for low time-scale factors with no reduction in the quality of output. Section 4 outlines the computational requirements of both the SOLA and the refined algorithm, the synchronised and adaptive overlap-add (SAOLA), and presents a comparison of their respective computational loads. Section 5 presents a comparison between SOLA and

SAOLA in terms of the quality of their output. Section 6 concludes the paper.

2. SYNCHRONISED OVERLAP-ADD (SOLA)

SOLA [1] segments the input signal x into m overlapping frames, of length N samples, each segment being S_a samples apart. S_a is the analysis step size. The time-scaled output y is synthesized by overlapping successive frames with each frame a distance of $S_s + k_m$ samples apart. 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

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

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_m is in the range $k_{min} \le k \le k_{max}$.

$R_m(k)$ is a normalized cross-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

$$y(mS_s + k + j) := (1-f(j))y(mS_s + k + j) + f(j)x(mS_a + j),\ 0 \le j \le L_m - 1 \quad (2a)$$

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

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

$$f(j) = 0,\ j < 0 \quad (3a)$$

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

$$f(j) = 1,\ j > L_m - 1 \quad (3c)$$

Typically, N is in the range of 20ms to 30ms (corresponding to 320 samples and 480 samples at a sampling rate of 16kHz, respectively), S_a is in the range of $N/3$ to $N/2$ samples, k_{min} is $-N/2$ and k_{max} is $N/2$. [2] and [3] report that k_{min} can be set to 0.

3. SYNCHRONISED AND ADAPTIVE OVERLAP-ADD (SAOLA)

As mentioned above, SOLA must determine the best overlap position from a range of possible overlap positions using a normalized cross-correlation function. The overlap range is an important consideration in SOLA's implementation, since too small a range results in a poor quality output and too large a range results in unnecessary computations. In an attempt to determine a 'reasonable' overlap range consider a periodic waveform overlapped with itself as shown in fig.1. The range shown in fig. 1 (a) is insufficient, since none of the possible overlap positions are strongly correlated, resulting in a poor quality output. Fig. 1 (b), however, displays an overlap range that would provide SOLA with an opportunity to locate a 'suitable', highly correlated overlap position, and thus produce a high quality output.

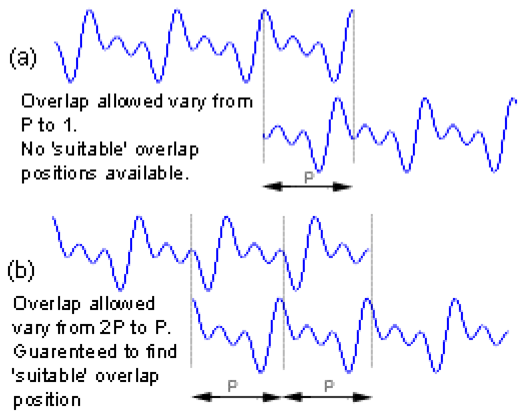
(a)
Overlap allowed vary from P to 1.
No 'suitable' overlap positions available.
P
(b)
Overlap allowed vary from 2P to P.
Guarenteed to find 'suitable' overlap position
P
P

Fig. 1 Overlapping periodic signals

For any perfectly periodic waveform an overlap range of $2P$ to P, where P is the period, is adequate, allowing the SOLA algorithm locate a highly correlated overlap position. However, for a speech signal, SOLA deals with various quasi-periodic segments of the input that have a wide range of periods. Therefore, in order to provide an adequate overlap range for all quasi-periodic segments we have found that the overlap should vary from $2P_L$ to 1, where P_L is the period of the lowest likely fundamental component.

P_L is typically in the range of 5ms to 8ms, and since N is typically in the range 20ms to 30ms we can approximate P_L to be $N/4$. Therefore the overlap should be allowed vary from $N/2$ to 1.
The overlap OL for the m^{th} iteration of the SOLA algorithm is given by

$$OL = (m-1)\,S_s + N + k_{m-1} - (mS_s + k_m) \quad (4a)$$
$$= N - S_s + k_{m-1} - k_m \quad (4b)$$

Since $S_s = \alpha S_a$ and letting $S_a = N/\beta$, OL is then given by

$$OL = N(\beta - \alpha)/\beta + k_{m-1} - k_m \quad (5)$$

To simplify following equations we let

$$Z = \beta/(\beta - \alpha) \quad (6)$$

Then

$$OL = N/Z + k_{m-1} - k_m \quad (7)$$

Since the overlap range for the m^{th} iteration is minimized for $k_{m-1} = k_{min}$ we must ensure that

$$N/Z + k_{min} - k_{min} \geq N/2 \quad (8)$$

and

$$N/Z + k_{min} - k_{max} \geq 1 \quad (9)$$

to physically ensure the desired overlap range exists. Also, in order to vary the overlap across the desired overlap range, the difference between k_{max} and k_{min} must be $N/2$ i.e

$$k_{max} - k_{min} = N/2 \quad (10)$$

To minimize the overlap range and satisfy both (8) and (9)

$$N/Z = N/2 + 1 \quad (11)$$

Since the derivation of the desired overlap range was not tightly constrained we can set

$$N/Z = N/2 \quad (12)$$

to simplify the following equations.
From (12) and (6)

$$2 = \beta/(\beta - \alpha) \quad (13a)$$

so

$$\beta = 2\alpha \quad (14)$$

Since $S_a = N/\beta$

$$S_a = N/(2\alpha) \quad (15)$$

Equation (15) provides us with an analysis step size that ensures that the overlap range is the optimal $N/2$ samples for all time-scale factors.

If S_a was fixed at N/γ, where γ is a fixed positive number, then the overlap range would be less than $N/2$ for $\alpha > \gamma/2$ when k_{m-1} is k_{min}. This can lead to a poor quality output since the possible overlap positions would be less than $N/2$. For $\alpha > \gamma$ there would be no possible overlap positions when k_{m-1} is k_{min}. For $\alpha < k_m/2$ the overlap would be greater than $N/2$, resulting in unnecessary computations. By allowing S_a adapt to the desired time-scale factor, as described by (15), an adequate overlap range is ensured for all time-scale factors and the number of computations required to provide a high quality output is minimized.

So far the only constraint that we have placed on k_{max} and k_{min} is that they differ by $N/2$ and that $k_{max} > k_{min}$, but to ensure that the overlap range physically exists for the first iteration of SOLA k_{min} must be greater than or equal to $-N/2$. Therefore, k_{max} must be greater than or equal to 0. k_{max} must also be less than or equal to $N/2$, to ensure the desired overlap range exists. Therefore, k_{min} must be less than or equal to 0. For our implementation k_{min} and k_{max} were set to 0 and $N/2$, respectively.

4. COMPUTATIONAL LOAD COMPARISON

The cross-correlation coefficient of SOLA may be efficiently computed using a simplified normalized cross-correlation coefficient given by

$$R_m(k) = \frac{\sum_{j=0}^{L_m-1} y(mS_s + k + j)\,x(mS_a + j)}{\sum_{j=0}^{L_m-1} |\,x(mS_a + j)\,| \sum_{j=0}^{L_m-1} |\,y(mS_s + k + j)\,|} \quad (16)$$

In general, for each iteration of the SOLA algorithm there is some fixed overlap, *F0*, and a search region *SR*. This is illustrated in fig. 2. Using the same approach set out in [4], each summation term of (16) initially requires *FO* additions and 1 extra addition for each step in the search range, resulting in 2(*FO* + *SR*) addition operations. The summation terms are also multiplied and then divided into the numerator for each step in the search range, resulting in 2*SR* multiply operations.

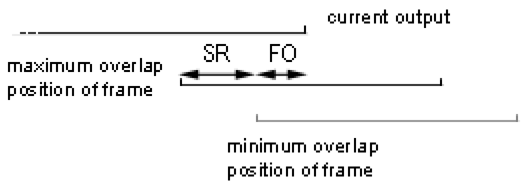
current output
maximum overlap position of frame
SR
FO
minimum overlap position of frame

Fig. 2. Search Range and Fixed Overlap

The numerator of (16) is a fraction of an *SR* + *FO* point convolution. [4] explains that the numerator can be efficiently calculated through the use of an FFT based fast convolution technique. Using the same steps as in [4] it can be shown that the total operations required to implement an *SR* + *FO* point convolution is 4(*SR* + *FO*)Log$_2$(*SR* + *FO*) + 4(*SR* + *FO*) multiplies and 6(*SR* + *FO*)Log$_2$(*SR* + *FO*) + 4(*SR* + *FO*) – 4 additions. However, SOLA is only concerned with a segment of length *SR* of the entire

$SR + FO$ convolution. Therefore only a fraction of the total convolution sequence is required. This fraction is given by $SR/(2(SR + FO))$, therefore, by assuming appropriate FFT pruning can be applied, calculation of the numerator then requires $2SR\mathrm{Log}_2(SR + FO) + 2SR$ multiplies and $3SR\mathrm{Log}_2(SR + FO) + 2SR - 2SR/(SR + FO)$ additions.

Having calculated $R_m(k)$ for each step in the search range, SR comparisons are then required to determine the maximum $R_m(k)$ and hence k_m.

Finally, linear cross-fading across the entire length of the overlapping regions requires $SR + FO$ additions and $2(SR + FO)$ multiply operations.

The total number of operations per iteration of the SOLA algorithm is then SR comparisons, $3(SR + FO) + 3SR\mathrm{Log}_2(SR + FO) + 2SR - 2SR/(SR + FO)$ additions and $4SR + 2SR\mathrm{Log}_2(SR + FO) + 2(SR + FO)$ multiplications.

For both SAOLA and SOLA the number of iterations required to time-scale a signal x of length L_x is equal to the number of analysis frames m, which is given by

$$m = L_x / S_a \qquad (17)$$

where S_a is the analysis step size. Letting $S_a = N/\beta$ and normalizing L_x to 1, table 1 shows the estimated number of computations required to implement SOLA.

	SOLA COMPUTATIONS
Multiplies	$[4SR + 2SR\mathrm{Log}_2(SR + FO) + 2(SR + FO)]\,\beta/N$
Additions	$[3(SR + FO) + 3SR\mathrm{Log}_2(SR + FO) + 2SR - 2SR/(SR + FO)]\,\beta/N$
Comparisons	$\beta SR/N$

Table 1. SOLA computational load estimate

We must now determine the typical values of SR and FO for various time-scale factors.

From (7), it can be seen that the average overlap will be N/Z, since the k values of (7) will, on average, cancel out. This overlap ensures that the output is time-scaled by α. From fig. 2, the overlap lies somewhere between FO and $FO + SR$. The average overlap will then also be given by $SR/2 + FO$. From the discussion above

$$N/Z = SR/2 + FO \qquad (18)$$

Again from (5), for $\alpha \leq \beta/2$ the overlap is guaranteed to be greater than or equal to $N/2$, the desired search range. Therefore, the search region SR is guaranteed to be $N/2$ for $\alpha \leq \beta/2$. Then from (18),

$$FO = N/Z - N/4 \qquad (19)$$

For $\alpha > \beta/2$ the overlap is no longer guaranteed to be greater than $N/2$. This has the effect of limiting the search range and reduces the average value for SR. The search range has a minimum range of N/Z and a maximum range of $N/2$, it is therefore reasonable to assume that a typical value for the search range lies midway between these values i.e.

$$SR = N/Z + (N/2 - N/Z)/2 \qquad (20a)$$

Then

$$SR/2 = [N/Z + (N/2 - N/Z)/2]/2 \qquad (20b)$$
$$= N/(4Z) + N/8 \qquad (20c)$$

From (18)

$$FO = N/Z - N/(4Z) - N/8 \qquad (21a)$$
$$= 3N/(4Z) - N/8 \qquad (21b)$$

From (21b) the average fixed overlap FO is negative for

$$Z > 6 \qquad (22)$$

Then using (5)

$$\beta/(\beta - \alpha) > 6 \qquad (23)$$

We will not consider the case where $\alpha > \beta$, since this case results in a search range of zero, resulting in a very poor quality output. From (23) FO is negative for

$$\alpha > 5\beta/6 \qquad (24)$$

Since the fixed overlap cannot physically be negative we will assume that (20a), (20b), (20c), (21a) and (21b) are valid for $\beta/2 < \alpha \leq 5\beta/6$. For $5\beta/6 < \alpha \leq \beta$ we will assume the fixed overlap is zero, therefore from (18)

$$SR/2 = N/Z \qquad (25)$$

then

$$SR = 2N/Z \qquad (26)$$

Table 2 displays an estimate of values for FO and SR for various time-scale factors.

α	**FO**	**SR**
$\alpha \leq \beta/2$	$N/Z - N/4$	$N/2$
$\beta/2 < \alpha \leq 5\beta/6$	$3N/(4Z) - N/8$	$N/Z + (N/2 - N/Z)/2$
$5\beta/6 < \alpha \leq \beta$	0	$2N/Z$

Table 2. SOLA fixed overlap and search range values.

Using the data from tables 1 and 2 and equation (6), the total computational load can be estimated for various time-scale factors and various values of β. As mentioned in [4], a digital signal processor (DSP) can perform single cycle multiply, add and compare operations. However, an application specific integrated circuit (ASIC) multiply operation is approximately equivalent to 16 addition operations. So, to calculate the total number of ASIC operations we weight the number of multiply operations by 16.

The computational load associated with SAOLA can be estimated by setting $\beta = 2\alpha$. Fig. 3 (a) and (b) show the ratio of SOLA computations to SAOLA computations with β, for SOLA, set to 2 and 3, respectively, for a selection of time-scale factors with N = 480 samples (corresponding to 30ms at a sampling rate of 16kHz) for a DSP implementation.

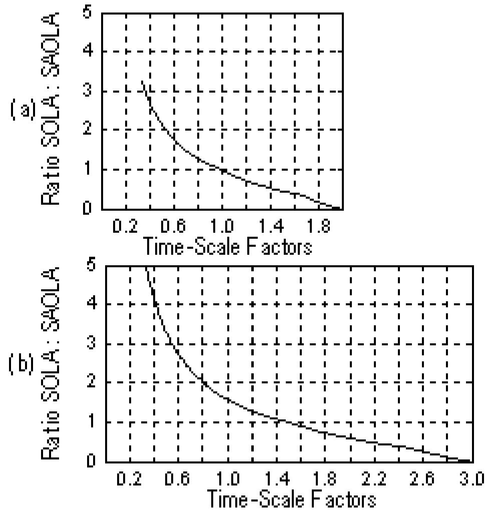

Fig. 3. Ratio of SOLA operations to SAOLA operations for (a) β = 2 and (b) β = 3

5. OUTPUT QUALITY COMPARISON

Eleven evaluation subjects of various age and gender carried out informal listening tests. The test comprised of 20 comparisons between a track time-scaled by SAOLA and the same track time-scaled by SOLA, using the same time-scale factor. The subjects were not informed which track was a SAOLA time-scaled track or which was a SOLA time-scaled track. The tests covered a selection of time-scale factors ranging from 0.5 to 3 and contained an equal number of male and female speakers. For all tests the sampling rate = 16kHz, N = 30ms, $k_{min} = 0$, $k_{max} = N/2$ and S_a for SOLA was fixed at $N/3$ i.e. $\beta = 3$.

The listening tests showed that for time-scale factors greater than $\beta/2$, i.e. 1.5, the output quality for speech signals time-scaled using SAOLA was better than the same signals time-scaled using SOLA. For time-scale factors less than or equal to 1.5 the output quality is approximately equal for both algorithms. A summary of the results is shown in table 3.

	$\alpha \le \beta/2$	$\alpha > \beta/2$
SAOLA much better than SOLA	1.5 %	33.1 %
SAOLA slightly better than SOLA	19.7 %	42.2 %
SAOLA equal to SOLA	62.1 %	11.1 %
SAOLA slightly worse than SOLA	16.7 %	11.7 %
SAOLA much worse than SOLA	0.0 %	1.9 %

Table 3. Summary of listening test results

6. CONCLUSION

A refinement to the commercially popular SOLA algorithm is introduced in which the analysis step size is allowed adapt to the desired time-scale factor. The refinement has the effect of improving the quality of the output for large time scale factors, at the expense of additional computational requirements, and reducing the number of computations required for low time-scale factors with no loss in the quality of the output.

The adaptive algorithm, SAOLA, is capable of producing comprehensible speech up to a factor of 8. Furthermore, SAOLA can take advantage of the computational savings outlined in [3].

7. REFERENCES

[1] 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.

[2] Hardam, E., "High quality time scale modification of speech signals using fast synchronised-overlap-add algorithms", Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing, Volume 1, pp. 409-412, 1990.

[3] Wong, J.W.C., Au, O.C. and Wong, P.H.W, **"**Fast time scale modification using envelope-matching technique (EM-TSM)**"**. Proc. of the IEEE International Symposium on Circuits and Systems, Volume 5, pp. 550–553, May 1998.

[4] Lawlor, B. and Fagan, A.D., "A Novel Efficient Algorithm for Audio Time-Scale Modification", Irish Signals and Systems Conference '99, National University of Ireland, Galway, 1999.