High Quality Time-Scale Modification of Speech using A Peak Alignment Overlap-Add Algorithm (PAOLA)

2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003.

David Dorran, Robert Lawlor and Eugene Coyle

教授:陳福坤 老師
學生:張育豪

Outline

1. Introduction
2. Synchronized Overlap Add (SOLA)
3. Peak Alignment Overlap Add (PAOLA)
4. Computational Load Comparison
5. Output Quality Comparison
6. Conclusion

Introduction

- TSM(Time Scale Modification)是一種可以讓聲音的速度變快或是變慢，而且音調(Pitch)不變的方法
- TSM的技術分成為:
  - 頻率域 (Phase Vocoder…etc)
  - 空間域 (OLA、SOLA、SAOLA…etc)
- 本篇作者將本篇方法PAOLA與自己提出過的方法SAOLA做比較

Synchronized Overlap Add (SOLA)

- SOLA是基於交相關函數找出分析音框x_m以及合成音框y中波形最相似的地方然後進行疊加動作

$$R[k]=\frac{\sum_{i=0}^{L-1} x[mS_a+i]\,y[mS_s+k+i]}{\sqrt{\sum_{i=0}^{L-1} x^2[mS_a+i]}\sqrt{\sum_{i=0}^{L-1} y^2[mS_s+k+i]}}$$

alpha >1 速度變慢，alpha<1 速度變快

N 音框長度

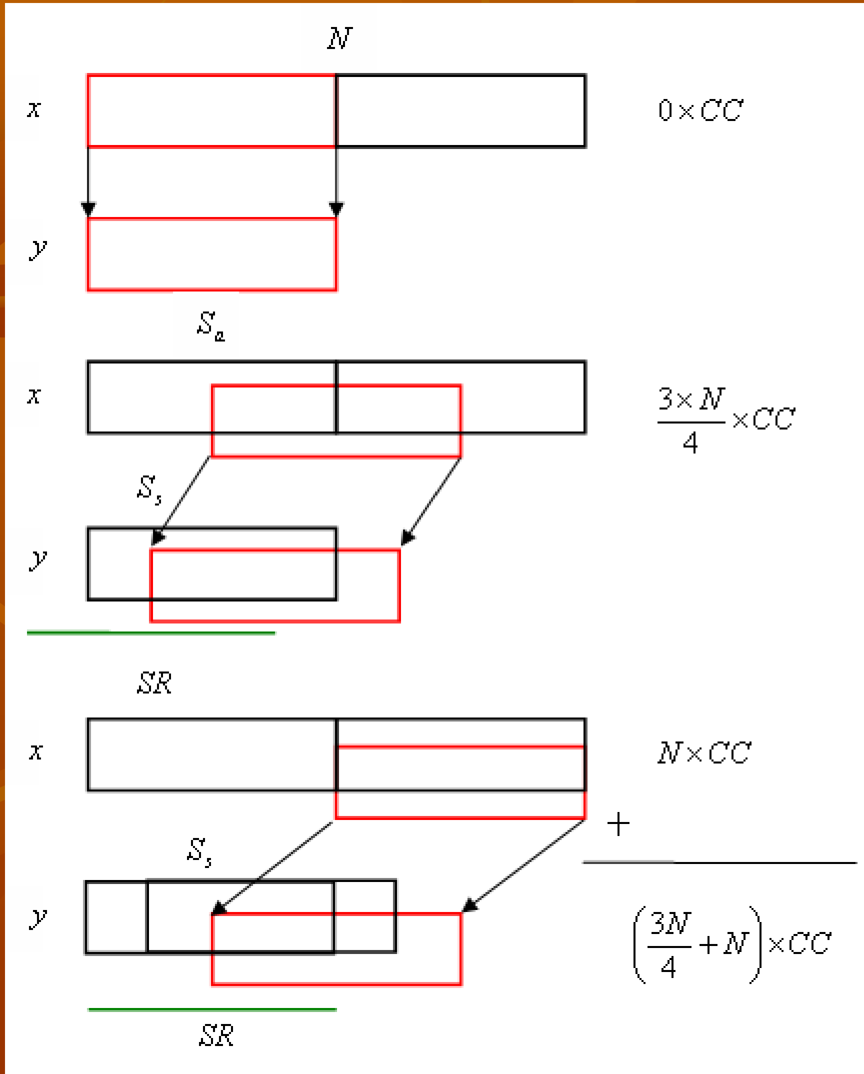
$S_a=N/2$ 分析音框長度

S_s=alpha*S_a 合成音框長度

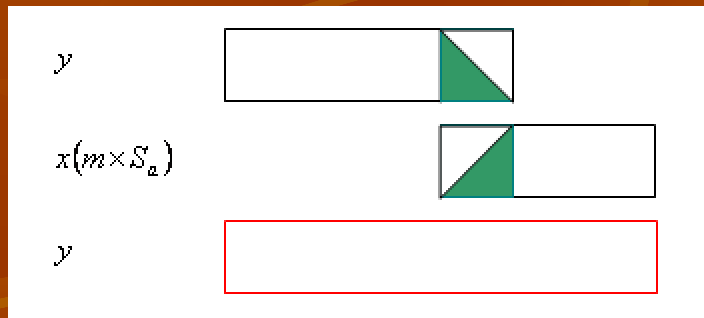
L爲疊加區域長度

k {-N/2,N/2} 搜尋範圍SR

Synchronized Overlap Add (SOLA)

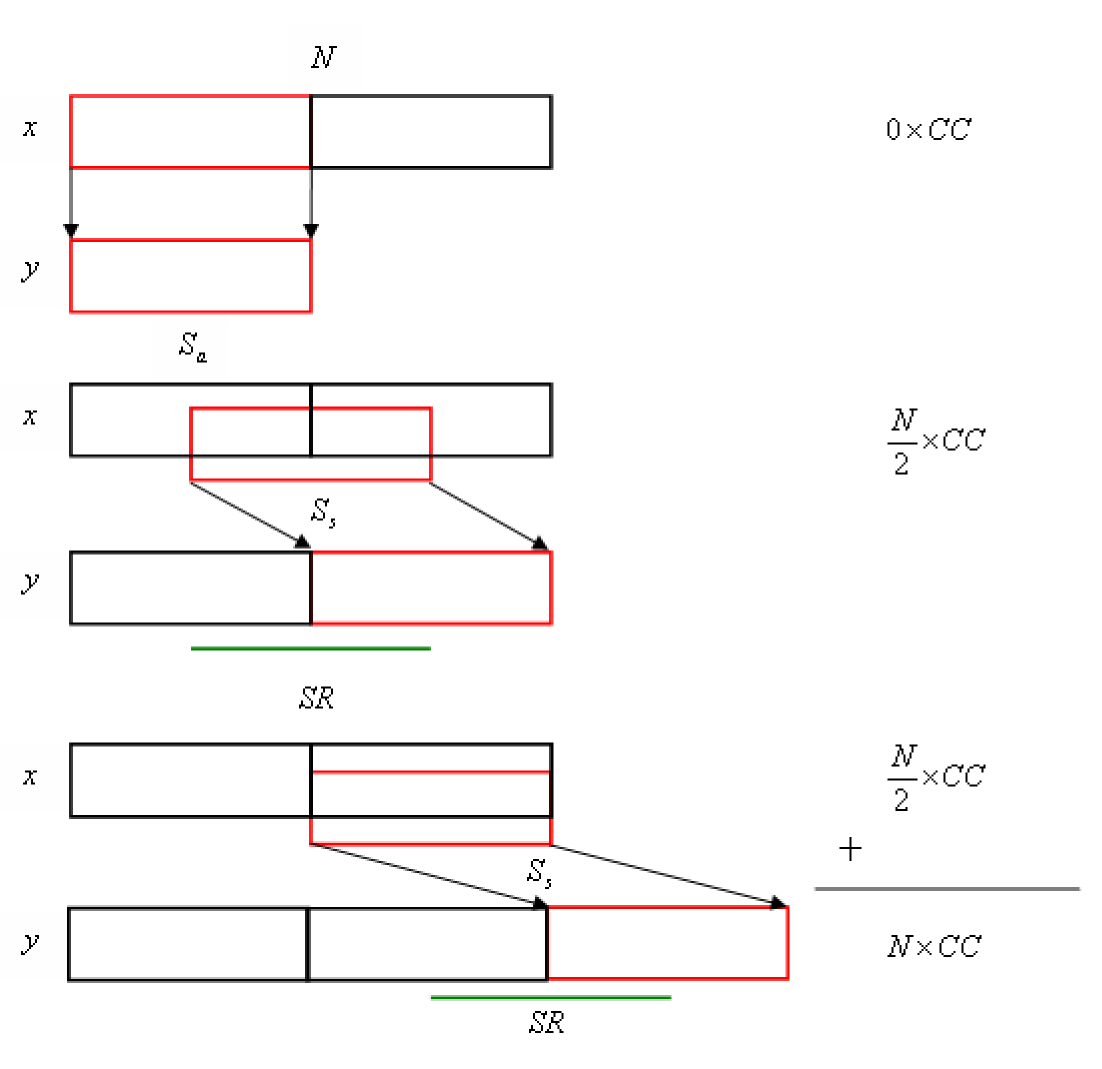


SOLA過程 alpha=0.5



疊加方式

Synchronized Overlap Add (SOLA)



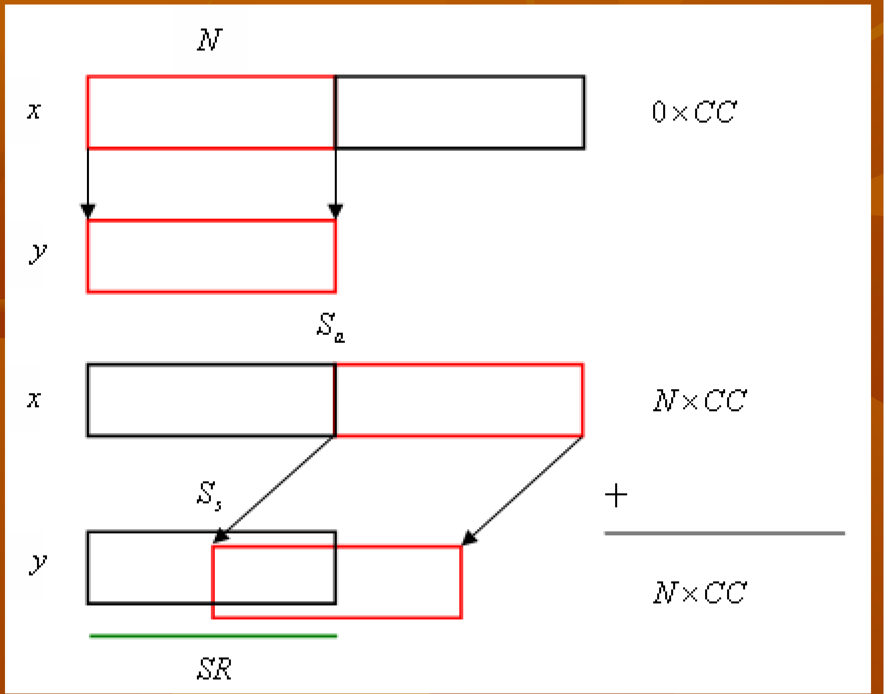
SOLA過程 alpha=2

Synchronized Overlap Add (SOLA)

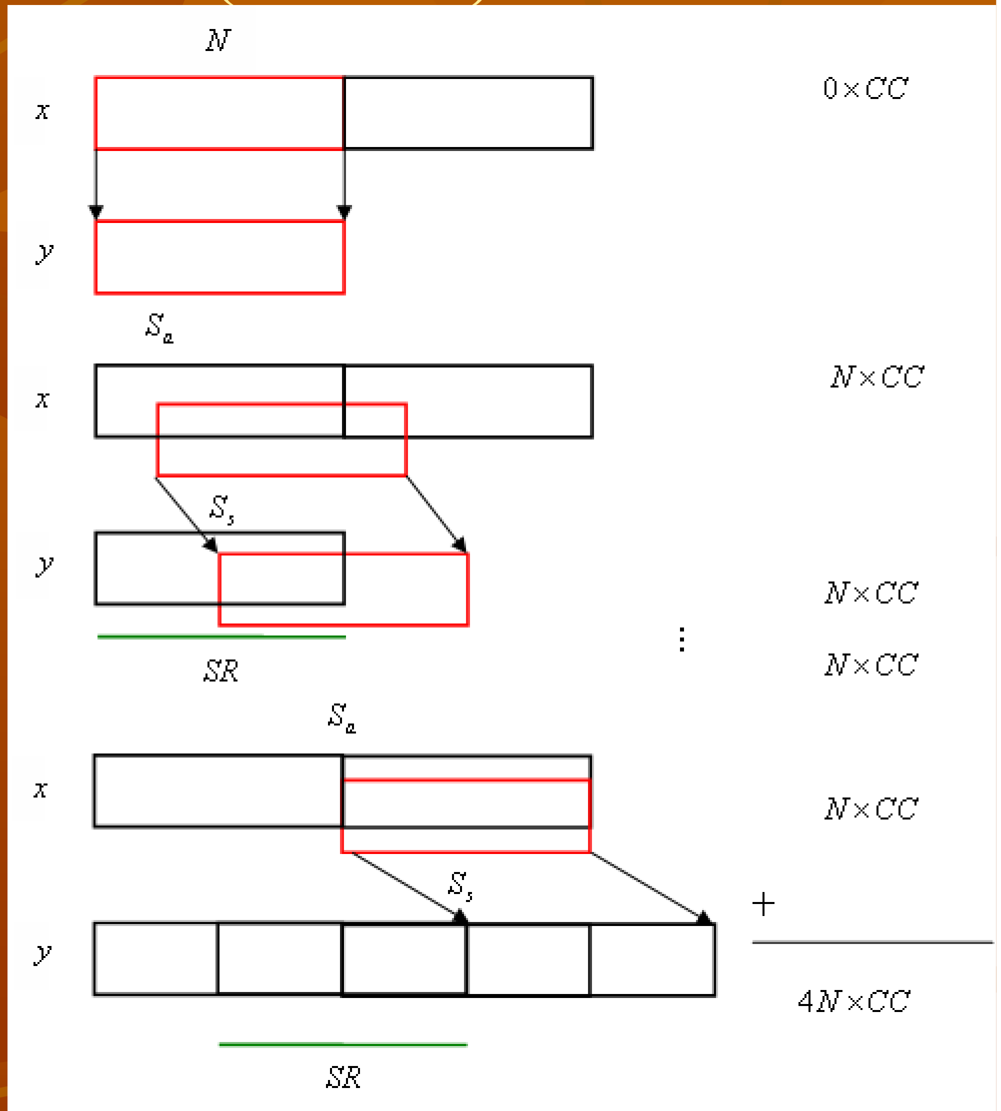
- 在SOLA中，參數N，S_a，以及搜尋範圍SR都是固定的
- 但是我們可以在SOLA計算中發現到其搜尋範圍是不固定的
- 所以作者提出了一個搜尋範圍固定，而且可以包含到SOLA所找到的值的方式，即稱爲SAOLA(Synchronized Adaptive Overlap Add)
- 作者將Sa的計算改成爲

$$S_a = \frac{N}{2 \times alpha}$$

Synchronized Overlap Add (SOLA)



SAOLA過程 alpha=0.5



SAOLA過程 alpha=2

Peak Alignment Overlap-Add Algorithm

1. 將訊號x切分爲$window$個長度N爲的分析音框x_m
   $0 \leqq m < n$。
2. 找出目前分析子音框x_m，搜尋從0到SR-1距離之間的最大波峰位置p_x。
3. 找出目前合成音框y的長度M_m，再往前SR距離之間找尋最大波峰位置p_y。
4. 最後將合成音框找尋到的最大波峰位置$y_m(p_y)$與分析子音框$x_m(p_x)$的依淡出淡入的加權方式進行疊加動作。

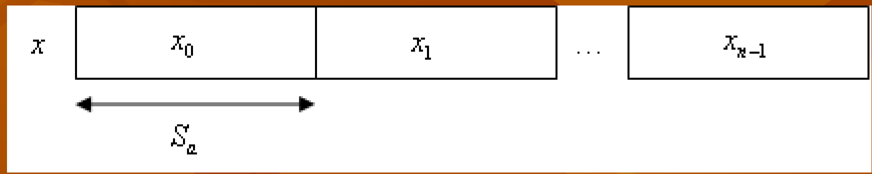
Peak Alignment Overlap-Add Algorithm

- 參數設定
  - Frame size N=240
  - alpha=0.5
  - S_a=N/(2 × alpha)=240
  - S_s=alpha × S_a=0.5×240=120
  - SR=N-S_s=120

Peak Alignment Overlap-Add Algorithm

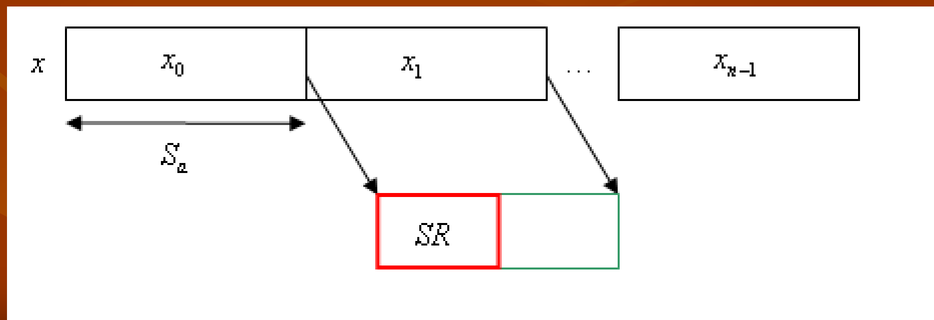
1. x_m is the m^{th} input frame and is given by

$$x_m = x(mS_a + j), \quad 0 \le j < N$$



S_a is the length of analysis frame
N is the length of frame

2. the maximum peak $x_m(p_x)$ is found in the region $x_m(j)$, $0 \leqq j < SR$,



Peak Alignment Overlap-Add Algorithm

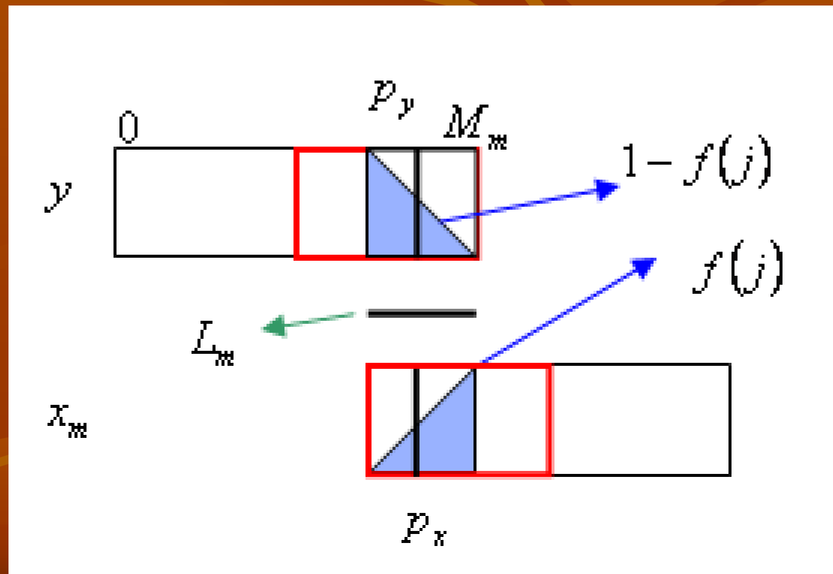
3. For the m^{th} iteration, the PAOLA algorithm first searches the current output for the maximum peak $y_m(p_y)$ in the region $y_m(M_m\text{-j}), 0 \leqq \text{j} < \text{SR}$

- M_m is the length of the current output y_m after m iterations
- SR is the length of the search region

Peak Alignment Overlap-Add Algorithm

4. The m^{th} input frame is then overlap-added with y_m such that the located peaks $x_m(p_x)$ and $y_m(p_y)$ are aligned producing y_{m+1}

$$L_m = p_x + M_m - p_y + 1$$



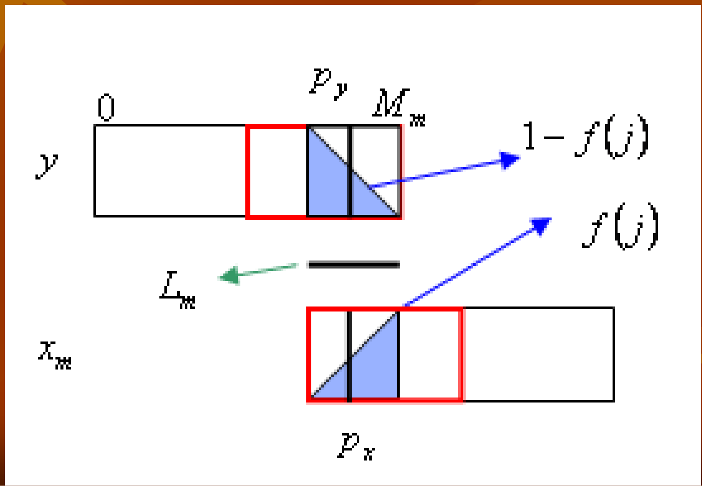
Peak Alignment Overlap-Add Algorithm

- The overlapping regions of y_m and the m^{th} input frame are weighted prior to combination resulting in

$$y_{m+1}(j) = y_m(j), \quad 0 \le j \le M_m - L_m - 1$$

$$y_{m+1}(M_m - L_m + j) = y_m(M_m - L_m + j) \times (1 - f(j)) + x_m(j) \times f(j), \quad 0 \le j \le L_m - 1$$

$$y_{m+1}(M_m - L_m + j) = x_m(j), \quad L_m \le j \le N$$



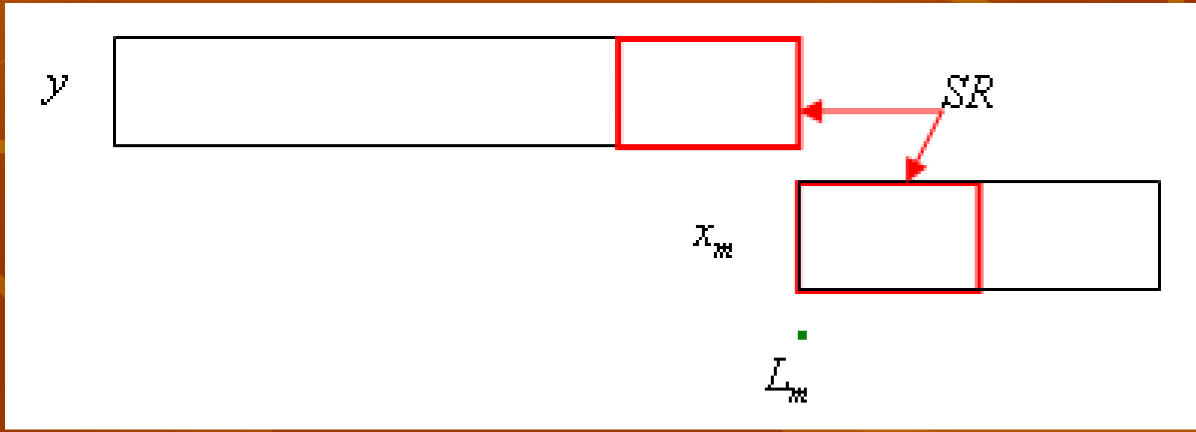
$$f(j) = \begin{cases} 0 & , j < 0 \\ \dfrac{j}{L_m - 1} & , 0 \le j \le L_m - 1 \\ 1 & , n > L_m - 1 \end{cases}$$

Peak Alignment Overlap-Add Algorithm

a. p_x=0 and p_y=M_m ,then

$$L_m = p_x + M_m - p_y + 1$$

$$L_m = 1$$



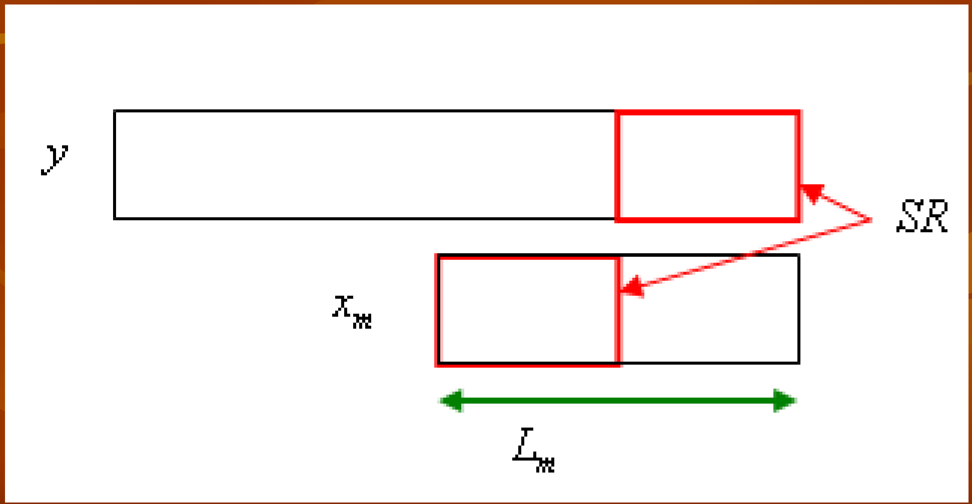
Peak Alignment Overlap-Add Algorithm

b. p_x=SR-1 and p_y=M_m-(SR-1) ,then L_m=2SR-1

$$L_m = p_x + M_m - p_y + 1$$

$$L_m = SR - 1 + M_m - (M_m - (SR - 1)) + 1$$

$$L_m = 2SR - 1$$



Peak Alignment Overlap-Add Algorithm

採用Sa=N/(2×*alpha*)的原因

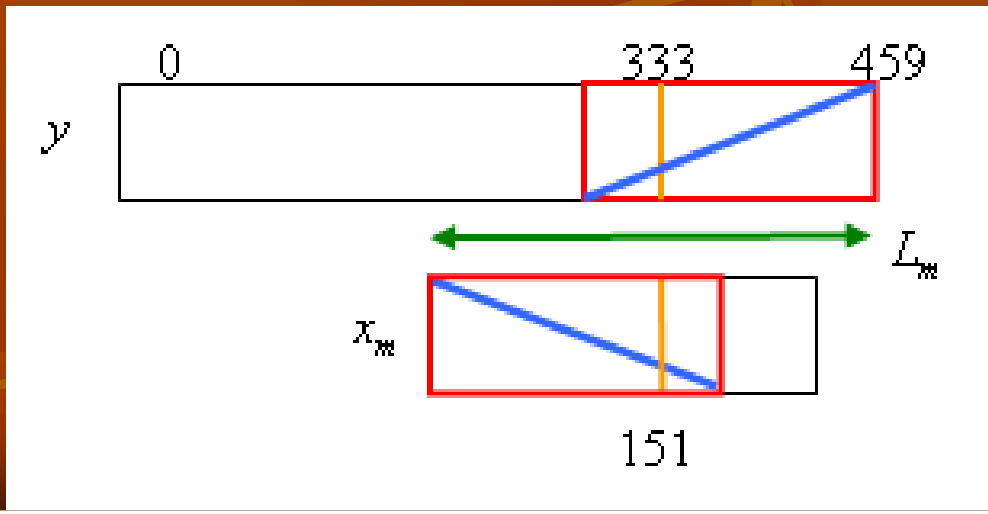
$S_a = N/2 = 120$

$S_s = alpha \times S_a = 0.5 \times 120 = 60$

$SR = N - S_s = 180$

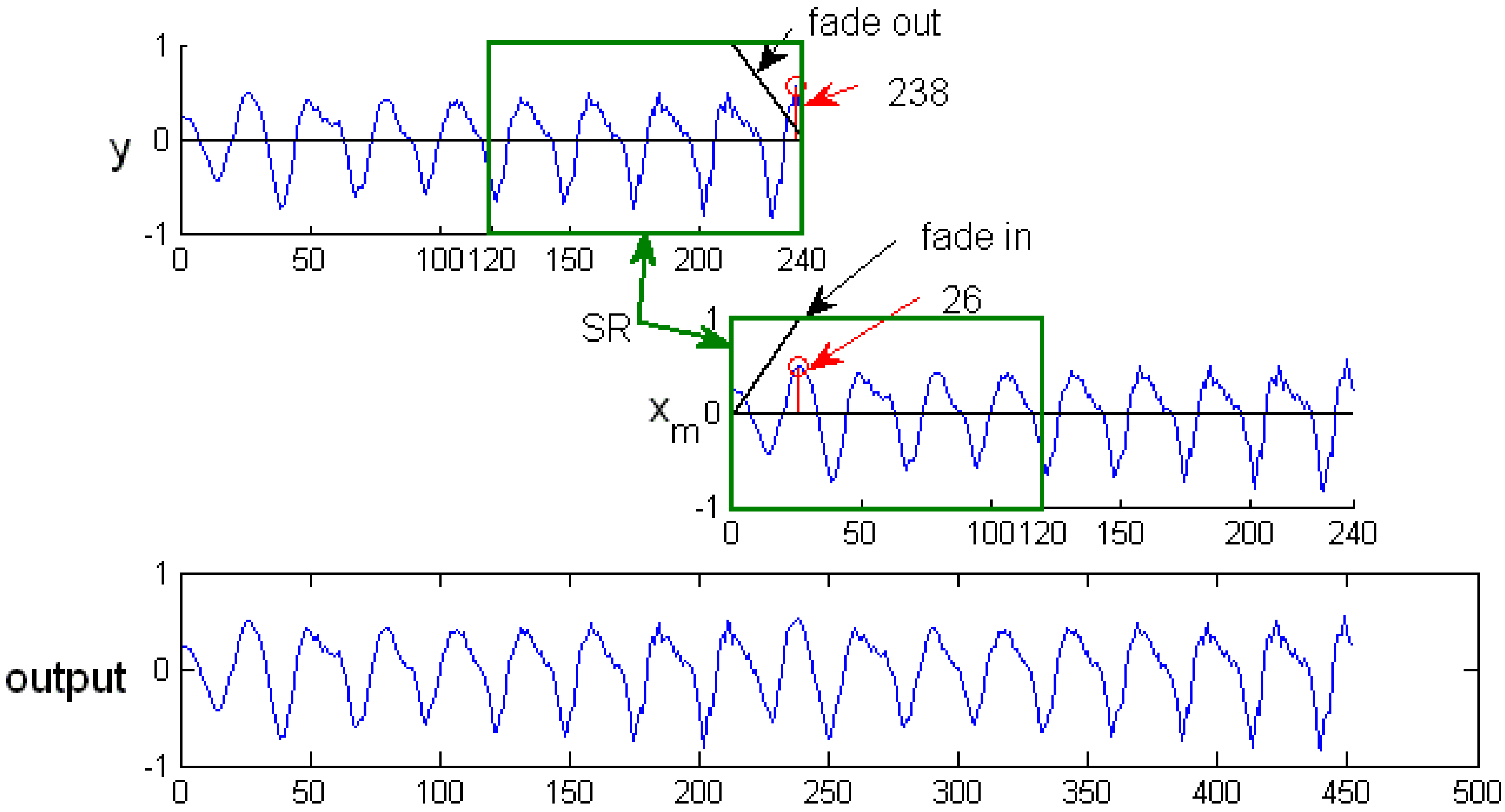
$M_m = 460, p_x = 152, p_y = 334$

$L_m = 152 + 460 - 334 + 1 = 279$



Peak Alignment Overlap-Add Algorithm

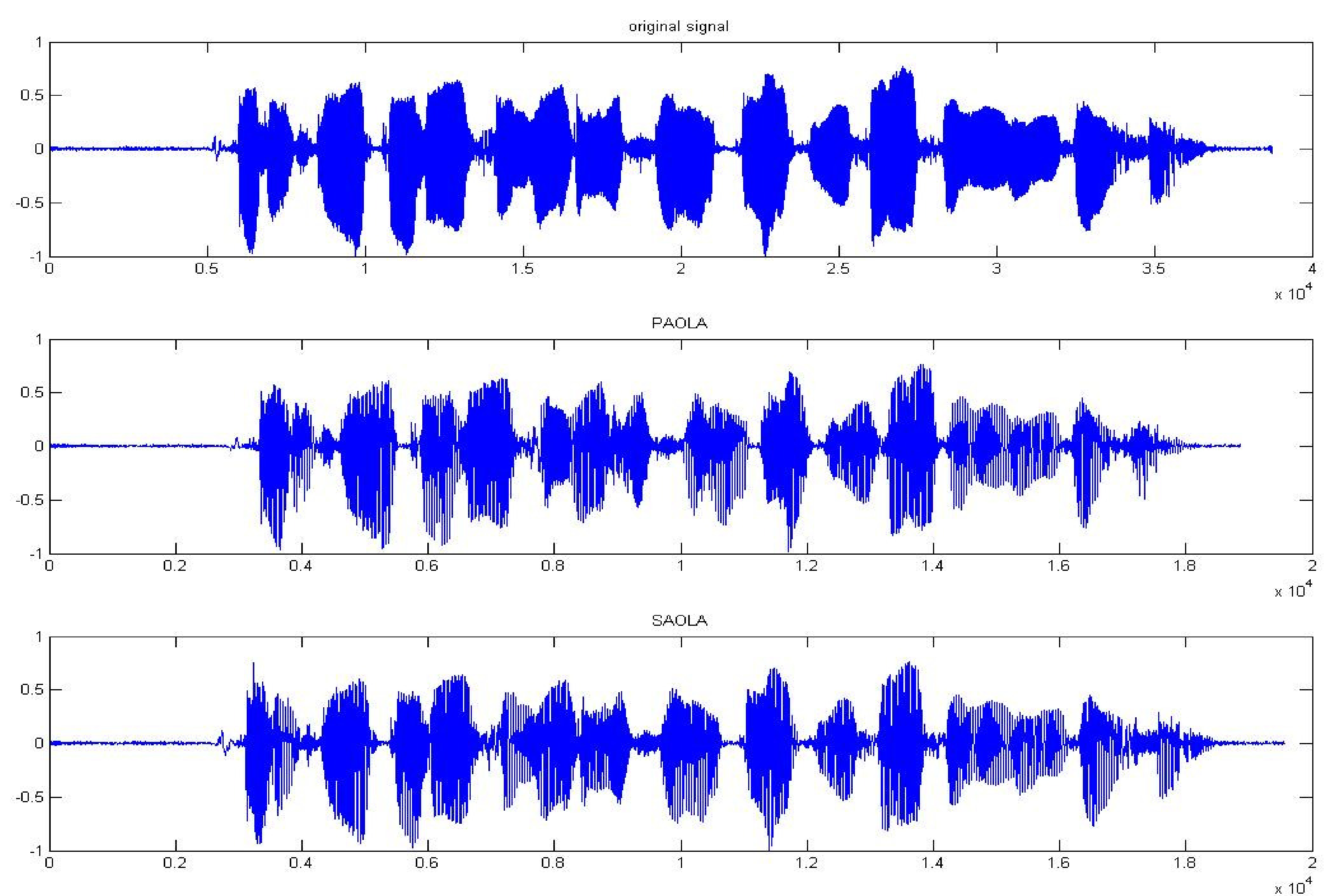
- M_m=240, p_x=26, p_y=238, L_m=29



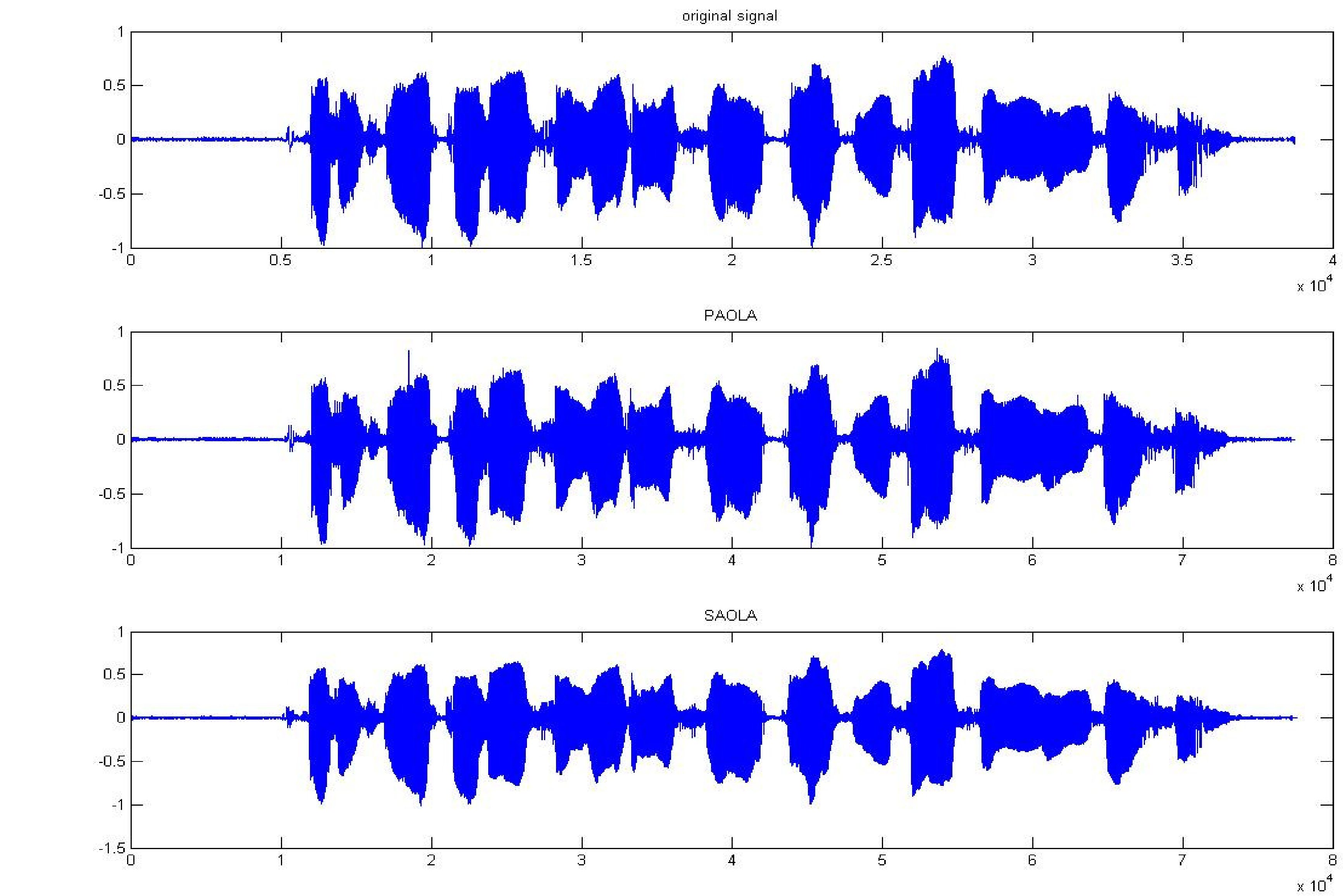
Computational Load Comparison

	SAOLA	PAOLA
Multiplies	$2\alpha\log_2(3N)+3\alpha$	$\left(\frac{\lvert 1-\alpha\rvert}{L_{stat}-SR}\right)2SR$
Additions	$3\alpha\log_2(3N)+\frac{\alpha}{2}+\frac{8\alpha}{3N}$	$\left(\frac{\lvert 1-\alpha\rvert}{L_{stat}-SR}\right)SR$
Comparisons	α	$\left(\frac{\lvert 1-\alpha\rvert}{L_{stat}-SR}\right)2SR$

Output Quality Comparison   alpha=0.5



Output Quality Comparison alpha=2



Conclusion

- PAOLA的語音品質相當接近於SAOLA的品質，且PAOLA在執行速度上比SAOLA快8倍