SPEECH SYNTHESIS USING PHASE VOCODER TECHNIQUES

JOSEPH DI MARTINO
Loria Laboratory
Université Henri Poincaré Nancy I
BP. 239; 54506 VANDOEUVRE FRANCE

Tel. (+33) 03 83 59 20 36 Fax: (+33) 03 83 41 30 79, E-mail: jdm@loria.fr

ABSTRACT

A new light is thrown on the Portnoff [1] speech signal time-scale modification algorithm. It is shown in particular that the Portnoff algorithm easily accommodates expansion factors bigger than 2 without causing reverberation nor chorusing. The modified Portnoff algorithm, which draws on spectral modification techniques due to Seneff [2], has been tested on several speech signals. The quality of the synthesized signal is totally satisfactory even for big expansion factors. The article gives a brief summary of the Portnoff algorithm and spells out the modifications introduced. It is shown that the phase unwrapping procedure constitutes a crucial point of the algorithm.

1. INTRODUCTION

Time-scale modification techniques are very useful in such fields as data transmission and assisting persons suffering from auditory deficiencies. We are convinced that in some cases of auditory deficiency, slowing down the speech signal can enhance the understanding of the vocal message. At the present, the PSOLA algorithm [3] remains the most popular algorithm for doing so. It is extremely easy to implement. It however relies crucially on the use of a good pitch detector. It is all well known that the use of a pitch detector undermines the perfect functioning of such an algorithm. This fact constitutes the main drawback of the PSOLA algorithm. This drawback is however doubly counterbalanced by the excellent quality of the synthesized signal and the record time in which it is obtained.

The interest of phase vocoder based algorithms resides in the absence of preliminary pitch calculations which is a major advantage. Moreover, the resulting synthesized signal is of very good quality even for big expansion factors. Such algorithms on the other hand, require very big computational power. For instance, the proposed modified Portnoff algorithm takes as much as 40 minutes to process 2 seconds of speech on a PC Pentium 166 for a time-scale expansion factor of 4 and around 20 minutes for the expansion factor of 2. This computational time involves no interpolation whatsoever of the DFT coefficient frames as proposed in the original contribution of Portnoff [1].

2. THE PORTNOFF ALGORITHM

The Portnoff algorithm can be summarized by the formula:

$$Y(n,\omega) = A(\beta n,\omega)\exp[j(\alpha(\beta n,\omega) + v(\beta n,\omega)/\beta)] \quad (1)$$

where $A(n,\omega)$ is the amplitude spectrum of the original signal, $\alpha(n,\omega)$ is a term which varies slowly with time and can be qualified as "phase modulator"; $v(n,\omega)$ is the unwrapped phase; β is the compression or the expansion factor and $Y(n,\omega)$, the spectrum of the synthesized signal.

Figure 1 gives the direct implementation of Eq. (1). It in fact schematizes the processing of one DFT channel. Three stages of processing can be distinguished from the figure. The first analyzes the speech signal. The second performs the transformation of the DFT coefficients, a stage during which the time-scale modification is realized. The third stage finally performs the synthesis in the strict sense of the word.

- The analysis stage preemphasizes the speech signal using a filter of the type $1 - \rho z^{-1}$. A Hamming window is next applied to the result of the analysis. The DFT coefficients are then calculated. A R:1 decimation operator finally operates on the sequence of DFT frames retaining one out of every R frames.

- The time-scale modification stage is divided into two parts in figure 1. The upper part is devoted to handling cases with expansion factor $\beta < 1$ while the lower part deals with the cases of $\beta > 1$. The case of $\beta < 1$ corresponds to expansion in the strict sense speaking while $\beta > 1$ on the contrary corresponds to compression. In the original algorithm, Portnoff chose β as a ratio of two integers D and I, *i.e.*, $\beta = \frac{D}{I}$.

SCHEMATIC DIAGRAM OF THE PORTNOFF ALGORITHM

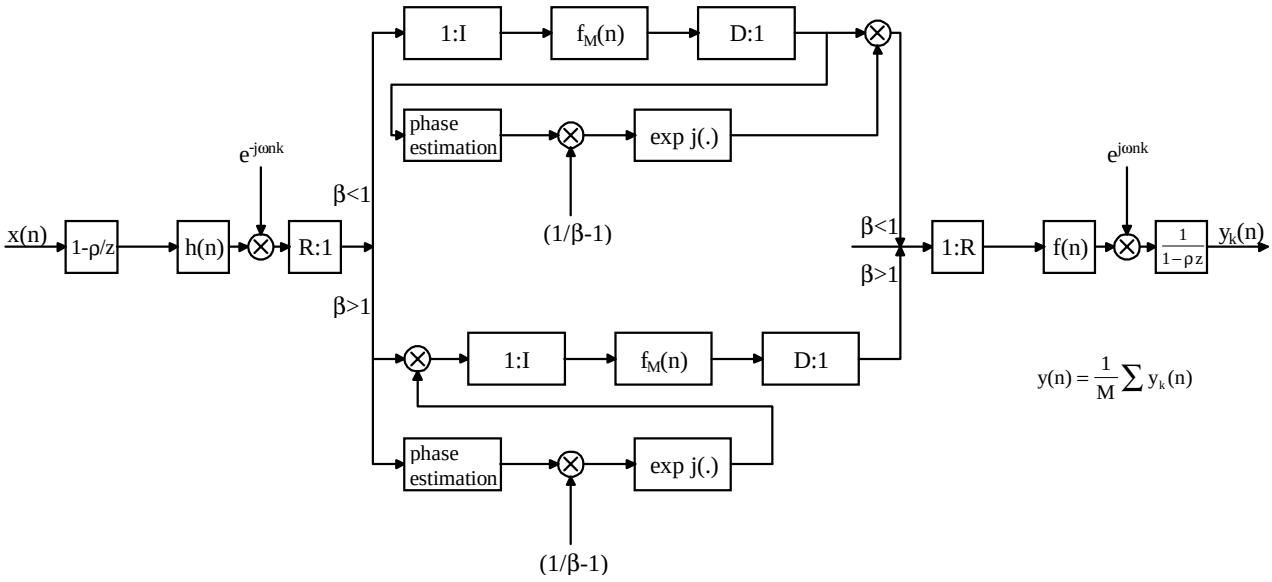


Fig. 1 - The *Portnoff algorithm*

Since D and I must be small, the set of the values of β is quite limited in size. The action of the zero-padding operator 1:I which inserts I-1 zeroes between two DFT coefficients belonging to the same frequency channel, followed by that of the interpolating filter $f_M(n)$ and the decimating operator D:1, results in the spectral level time-scale modification of the signal. The phase is multiplied by $1/\beta - 1$ instead of $1/\beta$ in order to avoid that the phase multiplication affects the phase modulation term.

- The synthesis stage is symmetrical in its action to the analysis stage as regards the decimation operation in that it restores all the DFT frames by the 1 :R and f(n) operators. The signals output by channels are all summed and de-emphasized.

Here follows the modifications we introduced into the original Portnoff algorithm:

1. The phase modulation term is assumed to be zero. This term is indeed barely visible when visualizing the unwrapped phases associated with the different DFT coefficients. S. Seneff [2] also ignored it in her algorithm.

2. The decimation operator R:1 has been suppressed from the analysis stage for two reason: 1) the duration of the calculation is not our primary concern; 2) our aim is to carry out a very accurate sample-to-sample phase unwrapping.

3. In an algorithm of this type based on phase vocoder techniques, very great care must taken in performing the phase unwrapping. In order to calculate the term $v(n,\omega)$, we implemented a phase unwrapping procedure which expresses $v(n,\omega)$ only in terms of $v(n-1,\omega)$, $\gamma(n-1,\omega)$ and $\gamma(n,\omega)$ where $\gamma(n,\omega)$ is the instantaneous phase of the DFT coefficient at the instant n and frequency ω. The next section will give the explicit calculation for $v(n,\omega)$.

4. As indicated by Portnoff [1] and Seneff [2], jumps of can occur at any moment, especially when the size of the DFT coefficient is close to zero. In this last case, the jump can assume any value with the behavior of the DFT coefficient becoming chaotic. The phase unwrapping procedure must take this into account. This is precisely what our algorithm does by considering a special case for the evaluation of the unwrapped phase when the modulus of the DFTcoefficient is below an experimentally determined threshold.

5. As regards the inevitable interpolation needed in the Portnoff algorithm to handle the time-scale modification itself, we replaced the Oetken filters [6] employed in the original Portnoff algorithm by Lagrange interpolating filters [5], resorting to a 24 point filter for the expansion factor 2 and to a 48 point filter for the expansion factor 4. The Oetken filters were difficult to synthesize.

SCHEMATIC DIAGRAM OF THE PROPOSED ALGORITHM

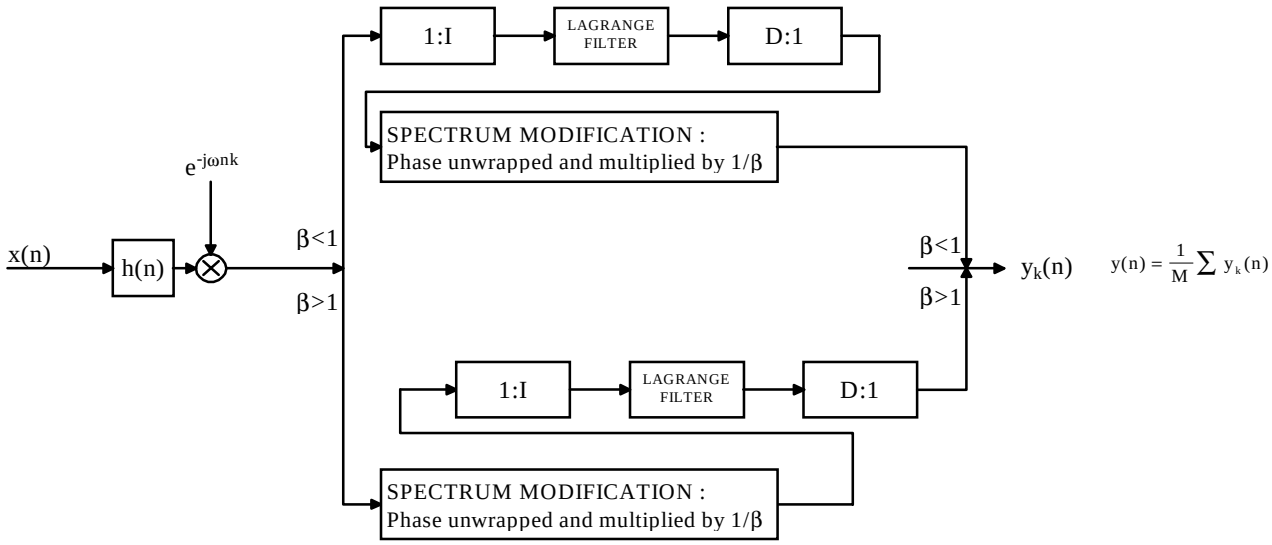


Fig. 2 - The *proposed algorithm*

The interest of Lagrange interpolating filters for this type of application resides in the facts that they are zero phase and preserve the support from which the interpolated points are calculated.

6.Contrary to Portnoff, we did away with the pre-emphasizing filter of the analysis stage on account of its non linearity phase response. The use of non linear phase filter in phase vocoder based applications can be a source of trouble.

Figure 2 schematically outlines the differences between the proposed algorithm and the original Portnoff algorithm [1].

3. THE PHASE UNWRAPPING PROCEDURE

Let $v(n,\omega)$ be the unwrapped phase at the instant n and frequency k and $\gamma(n,k)$ the corresponding instantaneous phase between 0 and 2π. The following cases were considered:

3.1. 2π jump : rising phase

1. Suppose that $\gamma(n-1,k) = 2\pi - \varepsilon_1$ where ε_1 is a small quantity. The phase jump under such a situation is $2\pi + \varepsilon$. What interests us is the quantity ε which is given by the equation :

$2\pi - \varepsilon_1 + 2\pi + \varepsilon = \gamma(n,k) = \varepsilon_2$.

In other words, $\varepsilon = \varepsilon_1 + \varepsilon_2$. Hence :

$$v(n,k) = v(n-1,k) + \left(2\pi - \gamma(n-1,k)\right) + \gamma(n,k)$$

2. Suppose $\gamma(n-1,k) = \varepsilon_1$. The phase jump in this case too is equal to $2\pi + \varepsilon$. ε is given by the equation:

$$\varepsilon_1 + 2\pi + \varepsilon = \gamma(n,k) = 2\pi - \varepsilon_2$$

from which one derives: $\varepsilon = -\varepsilon_1 - \varepsilon_2$. Hence :

$$v(n,k) = v(n-1,k) - \left(2\pi - \gamma[n][k]\right) - \gamma\left(n-1,k\right)$$

3.2. π jump : rising phase

1. Supposing $\gamma(n-1,k) = 2\pi - \varepsilon_1$. In this case the phase jumps is equal to $\pi + \varepsilon$ where ε is given by the equation:

$2\pi - \varepsilon_1 + \pi + \varepsilon = \gamma(n,k) = \pi + \varepsilon_2$ from which we deduce $\varepsilon = \varepsilon_1 + \varepsilon_2$. Consequently :

$$v(n,k) = v(n-1,k) + \left(2\pi - \gamma(n-1,k)\right) + \left(\gamma(n,k) - \pi\right)$$

2. Supposing $\gamma(n-1,k) = \varepsilon_1$. The phase jump is still given by $\pi + \varepsilon$. But this time, ε is given by the equation:

$$\varepsilon_1 + \pi + \varepsilon = \pi + \varepsilon_2$$

from which one deduces $\varepsilon = \varepsilon_2 - \varepsilon_1$ which leads to $v(n,k) = v(n-1,k) - \gamma(n-1,k) + \left(\gamma(n,k) - \pi\right)$.

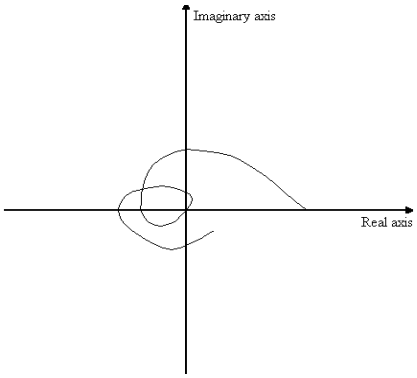
3.3. Descending phase jump

$v(n,k)$ in this case is given by: $v(n,k) = -v(n,N-k)$, where N is the number of the DFT coefficients used.

3.4. Chaotic phase jumps

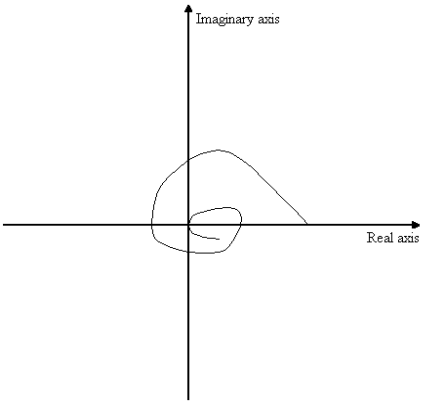
A chaotic phase jump can occur when the modulus of the DFT coefficient approaches zero. Figures 3, 4 and 5 illustrate this phenomenon. To deal with situations like this, we update the

unwrapped phase at the instant *n* by taking the previous phase as the current phase: $v(n,k) = v(n-1,k)$ to avoid the chaotic phase jump. This constitutes the third modification introduced into the Portnoff algorithm as previously stated in section 2 above.



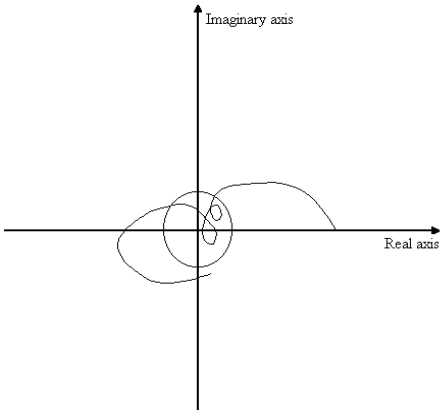
In this case, the DFT coefficient makes a π jump

Fig. 3 - *A π jump.*



In this case, the DFT coefficient makes a 2π jump

Fig. 4 - *A 2π jump.*



A chaotic phase jump was observed in the region close to the origin. The phase unwrapping procedure must take this into account

Fig. 5- *A chaotic jump*

4. RESULTS

We were able to slow down several speech signals of both female and male voices using as expansion factors 2, 3, 4 and even 5. The results obtained are very satisfactory. The resulting synthesized sound files were practically devoid of any reverberation noise. The chorusing effect reported in [4] was also totally absent from the files even in the cases of big expansion factors such as 4 and 5. The intelligibility of the synthesized sound was perfectly maintained as well as the characteristics of the speaker. Finally, the synthesized sound is of a very good quality. The interested reader could verify our assertions by listening to the different synthesized sound files. The sentence, uttered by a female speaker, is: "The quick fox jumps over the lazy dog" [sound A0442S01.WAV]. The sound files [sound A0442S02.WAV] and [sound A0442S03.WAV] contain the same sentence but slowed down by a factor of 2 ($\beta = 0.5$) for the one, and by 3 $\beta = 0.33$ for the other. The proposed modified Portnoff algorithm is also capable of compressing speech signals. The sound file [sound A0442S04.WAV] is a compressed version of the original sentence using a factor of 0.66, *i.e.*, $\beta = 1.5$.

5. CONCLUSION

This article proposes a modification of the Portnoff algorithm[1] inspired by the spectral modification techniques proposed by S. Seneff [2]. The results are very satisfactory even for big expansion factors. In the not distant future, we intend comparing our algorithm with PSOLA. The results of this comparison will form the topic of a future publication.

6. REFERENCES

[1] M.R. Portnoff "Time Scale Modification of Speech based on Short time Fourier Analysis“, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-29, No. 3, June 1981.

[2] S. Seneff ”System to Independently Modify Excitation and/or Spectrum of Speech Waveform Without Explicit Pitch Extraction“, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-30, No. 4 August 1982.

[3] E. Moulines, F. Charpentier “Pitch-Synchronous Processing Techniques for Text-To-Speech Synthesis Using Diphones”, Speech Communication 1990, pp. 453-467.

[4] E. Moulines, Jean Laroche “Non-Parametric Techniques for Pitch-Scale and Time-Scale Modifications of Speech”, Speech Communication 16, 1995, pp.175-205.

[5] R.W. Schaffer, L.R. Rabiner “A Digital Signal Processing Approach To Interpolation”, Proceedings of the IEEE, vol. 61, No. 6, pp. 692-702, June 1973.

[6] G. Oetken, T.W. Parks, H.W. Schussler, “New Results In The Design Of Digital Interpolators”, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-23, No. 3, pp. 301-309, June 1975.