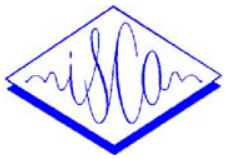

EUROSPEECH '97
5th European Conference on Speech
Communication and Technology
Rhodes, Greece, September 22-25, 1997

HIGH RESOLUTION PROSODY MODIFICATION FOR SPEECH SYNTHESIS

Francisco M. Gimenez de los Galanes and David Talkin
Entropic Research Laboratory, Inc.
600 Pensylvannia Ave. SE, Suite 202. Washington, DC. 20003
Tel. +1 202 547 1420, FAX: +1 202 546 6648, E-mail: galanes@entropic.com

10.21437/Eurospeech.1997-208

ABSTRACT

In this paper we will introduce RTIPS, a system for arbitrary high-resolution modification of the prosodic variables of speech: fundamental frequency, rhythm (segmental duration) and intensity. It is based on the Resample and ovelap-add (R-OLA) algorithm for fundamental frequency and duration modification of speech. The algorithm works pitch-synchronously in order to accurately modify the pitch contour, and it uses estimates of the glottal closure instants (epochs) as the synchronism marks. This technique is very similar to other OLA-based methods for time or pitch modification, but because of the introduction of the resampling step, voice quality (especially for high-pitched voices) is much more natural after resynthesis, at any given output sampling frequency. The reliability of the R-OLA algorithm is highly dependent on the accuracy of the method used for epoch detection, so this preprocessing step has to be carefully designed.

1. ALGORITHM DESCRIPTION

The basic R-OLA algorithm takes as input the raw speech samples and an estimate of the glottal closure instants (epochs). By integration of the desired fundamental frequency contour, a set of synthetic epochs is generated. The differential characteristic with former methods is that these synthetic epochs are not quantized, representing the desired F0 with full accuracy. The algorithm will generate a sequence of frames using these epochs as synchronism marks, and obtain the synthetic signal by overlap-add. The overlapping synthesis frames are generated by framing the original signal in a pitch-synchronous mode with a rectangular window, shifting these frames to the nearest previous sampling interval of the synthetic epoch and resampling these frames to fine-adjust the glottal closure instant to the desired time instant. The frames so generated and synchronized are then windowed using a non-symmetrical window and added, in a manner similar to [1][2][3].

The synthesis resampling step is needed because processing high pitched voices at a relatively low sampling frequency (16 kHz or less) can result in a low quality synthetic speech due to the implicit F0 quantization. A straightforward solution to this problem is processing an upsampled version of the speech, which either means extra storage or extra computation (the amount of extra computation depends on the ratio between the original and target sampling frequencies, and it can be significative since for high resolution pitch adjusting, a sampling rate higher than 44.1 Khz can be needed.) The approach taken in this work is to use an interpolator to adjust the delay of every windowed portion of speech. The design of this interpolator is so that it can be computed for every arbitrary delay on the fly, with a modest amount of computation, making it suitable for real-time applications.

1. 1. Analysis epoch computation

In order for the synthetic signal to accurately follow the desired F0 contour, high-resolution epoch detection is needed. One possibility is to use any standard method for epoch detection, but operate on an upsampled version of the original signal, as we did for our detector. The approach taken was to use two sources of information, namely F0 detected using crosscorrelation and dynamic programming [4] and peak-picking the integrated LPC residual. These two sources are combined to generate an extremely accurate estimate of the glottal closure instants. This step is performed only once, and the results are saved for further processing.

1. 2. Joint fundamental frequency and time-scale modification

The system under discussion takes as input the desired fundamental frequency contour and the desired time-warping function. By iteratively integrating these two functions, a set of synthetic epoch locations is generated. These locations are not quantized to the signal sampling frequency intervals or to any other interval than the accuracy permitted by the arithmetic processor. Therefore, they closely represent the desired prosodic contour and do not introduce artifacts in the form of artificial jitter or harmonic mix-up.

The time-warping function establishes a projection of the original and the synthetic time axes that is used to determine which segment of the original waveform is mapped into what time at the synthesis axis. This correspondence

is implemented at the frame level. When the combination of fundamental frequency and time-scale modification implies a higher density of epochs, frames will be repeated in order to make for this higher frame rate. Similarly, when the modification implies a sparser set of pitch marks, selected frames will be deleted (not used).

Once the frame correspondence is established, the synthesis frames are generated as follows:

a) A portion of the original signal is selected by applying a rectangular window around the synchronism epoch mark.The segment selected is the portion included between the previous epoch location and the next epoch location (the frame is exactly two periods long). This is the analysis frame.

b) The selected frame is shifted to the closest sampling interval that is previous to the synchronism mark.

c) Using a fast resampling method described below, the frame is fine-shifted to the exact position (time) as requested by the synthetic epoch mark.

d) In order to avoid reverberation, some windowing scheme is needed. The approach taken is the following:

Consider two consecutive synthesis frames, at synthesis indexes i-1 and i, and also consider the two periods of each synthesis frame separately. For the first period of frame at index i, there is some overlap with the second period of the previous frame, at index i-1. The length of the synthetic period is Ps(i). The length of the first period in the frame is P_1 (i), and the length of the second period in the previous frame is P_2(i-1). The window chosen is half a Hanning window for both periods (at i-1 and i). For the second period at i-1, it is a decreasing window (from 1 to 0) of length min(Ps(i), P_2(i-1). For the first period at i, the window is the raising part (from 0 to 1) of length min(Ps(i), P_1(i)).

This scheme implies that every frame is windowed using a non-symmetrical window, formed by two consecutive sections: the first half of a Hanning window with length equal to min(Ps(i), P_1(i)) followed by the second half of a Hanning window of length min (Ps(i+1), P_2(i)). This strategy minimizes the distortion introduced by the windowing step (for instance, if no prosodic change is applied, the synthetic signal is perfectly equal to the original signal.)

e) Intensity is controled by multiplying the synthesis frames by a gain factor.

f) The last step is the addition of the overlapping frames to generate the synthetic signal.

2. RESAMPLING METHOD

A scheme designed to overcome the problems associated with the standard resampling chain is presented below, in the context of signal reconstruction and resampling.

2. 1. Non Integer delay

Some applications require introducing a non integer delay in the signal. The common approach to this problem is to reconstruct the original signal from its samples and sample it again introducing the desired delay in the analog domain [5].

The whole process can be performed digitally by upsampling the digital signal, applying a digital "reconstruction" filter at that much higher sampling rate, introducing an integer delay and downsampling, where the admissible quantization of the delay gives us the minimum sampling frequency at which to operate.

2. 2. Optimization for non-integer unquantized delay

This delay chain can be optimized for variable, non-quantized delays, by looking at the modifying equations:

The reconstructed signal is

$$x(t) = \sum_{n=-\infty}^{\infty} x[n]\,\mathrm{sinc}\left(\frac{\pi(t-nT_s)}{T_s}\right)$$

and when resampled again after introducing a small delay, we get

$$y[m] = \sum_{n=-\infty}^{\infty} x[n]\,\mathrm{sinc}\,\pi f_s[(mT_s-\delta)-nT_s]$$

$$y[m] = \sum_{n=-\infty}^{\infty} x[n]\,\mathrm{sinc}\,\pi f_s[(m-n)T_s-\delta]$$

Expanding the *sinc* function as *sin(x)/x*, the *sin* can be rewritten as

$$\sin\pi f_s[(m-n)T_s-\delta] =$$
$$\cos\pi f_s\delta\sin\pi(m-n)-\sin\pi f_s\delta\cos\pi(m-n)$$

but $\cos\pi(m-n)=\pm1$, $\sin\pi(m-n)=0$ so

$$y[m] = \sum_{n=\infty}^{\infty} x[n]\frac{(-1)^{(m-n)}\sin\pi f_s}{\pi f_s[(m-n)f_s-\delta]}$$

where the *sin* function has a constant argument. If $0<\delta<T_s$, we can define $\delta<\alpha T_s$, where $0<\alpha<1$. Then we obtain

$$y[m] = \sum_{n=-\infty}^{\infty} (-1)^{(m-n)} x[n] \left(\frac{\sin \alpha\pi}{\pi}\right) \frac{1}{(m-n)+\alpha}$$

In real applications the limits of the sum can not be infinite, and must be reduced to a sensible integer value, introducing some distortion in the resulting signal. This distortion can be minimized by multiplying the resulting filter coefficients by a tapered window as in [6].

2. 3. Synthesis strategy

RTIPS assumes that the input speech is sampled at a relatively low sampling frequency, but the epoch locations have been estimated with a higher resolution. This is the case shown in Fig 1. where the epoch mark does not correspond to any actual sample.

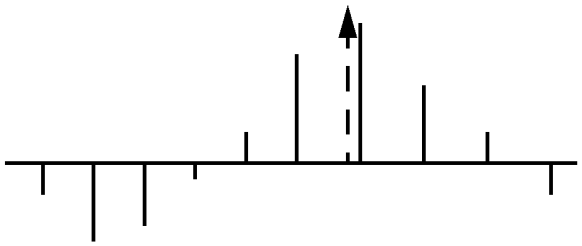

Fig 1. Samples from a voiced speech segment, in an environment of the glottal closure. The dashed arrow shows the real glottal closure instant.

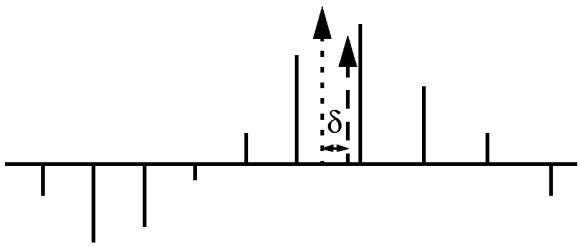

Fig 2. After F0 modification, the synthetic epoch is delayed with respect to the original epoch by a non-integral factor δ.

When modifying F0, the synthetic epoch locations will not generally correspond with the analysis epoch locations. I.e., we will find a situation as depicted in Fig 2. In this case, if we use the signal without the resampling step, extra jitter will be generated. In order to correct this phenomenon, we can introduce a non-integral delay in the short term signal, using the resampling approach. Note that this delay can be either positive or negative. Thus, the non-integral analysis and synthesis dealys can be combined and implemented in a single resampling step.

3. LP RTIPS ALGORITHM

The basic RTIPS algorithm can be applyed to the residual signal after LPC analysis, in the same manner as we did in the standard case modifying the real speech signal. Once the prosodics have been adjusted, an LPC synthesis filter is applied to this synthetic residual, generating the output speech. The main issue in LP-RTIPS is the synchronization of the LPC analysis-synthesis stage with the RTIPS algorithm, in order to preserve the spectral characteristics of the original speech in the scaled/warped version. Therefore, the LPC analysis is also pitch-synchronous: for every pitch mark, we also generate a linear prediction coefficients set.The analysis filtering is performed in an environment of the pitch mark (typically from the center of the left period to the center of the right period). By duplicating/eliminating LPC coefficient sets in a totally similar manner as we do with the RTIPS frames, we ensure that the additional distortion added will be minimum [7].

4. RESULTS AND CONCLUSIONS

Figure 3 shows the spectrograms of a segment of natural speech (top) and the corresponding pitch-modified version using two different methods: standard PSOLA (center) and RTIPS (bottom). The most noticeable differences in this case are around times 1038 and 1038.45, where the harmonic structure, specially in the range 1500-4000Hz is better preserved by RTIPS. This is a typical result: the increased resolution of the method minimizes the introduction of additional jitter during synthesis.

While the system as introduced here has been built to modify the prosody of prerecorded messages, it can also be applied to speech synthesis with very small modifications: the system would access a table of prerecorded segments (basic units), that can be concatenated to construct any desired message. The system described here would perform the changes needed to adjust the original prosody of the recorded units to the synthesis prosody as a previous step to concatenation.

5. REFERENCES

[1]Moulines, E., F. Charpentier. 1990. "Pitch-synchronous waveform processing techniques for test-to-speech using diphones." *Speech Communication, Vol 9, No.5-6. Amsterdam.*

[2] Crochiere, R.E. 1980. "A weighted overlap-add method of short-time Fourier analysis/synthesis." *IEEE Trans. on ASSP, Vol ASSP-28, No. 1:99-102.*

[3] Griffin, D.W., J.S. Lim. 1984. "Signal estimation from modified short-time Fourier transform." I*EEE Trans. on ASSP, Vol 32, No. 2: 236-243.*

[4] Talkin, David. 1995. "A robust Algorithm for Pitch Tracking (RAPT)", in *Speech Coding and Synthesis*, by Kleijn and Paliwal (Editors), Elsevier, Amsterdam.

[5] Oppenheim, A.V., R.W. Schafer. 1989. *Discrete-time signal processing*. Prentice Hall, Englewood Cliffs, N.Y.

[6] Gimenez de los Galanes, F.M., M.H. Savoji, J.M. Pardo. 1995. "Speech synthesis system based on a variable decimation/interpolation factor." *IEEE Proc. ICASSP'95. Detroit.*

[7]Gimenez de los Galanes, F.M., M.H. Savoji, J.M. Pardo. 1994. "New algorithm for spectral smoothing and envelope modification for LP-PSOLA synthesis." *IEEE Proc. ICASSP'94. Adelaide.*

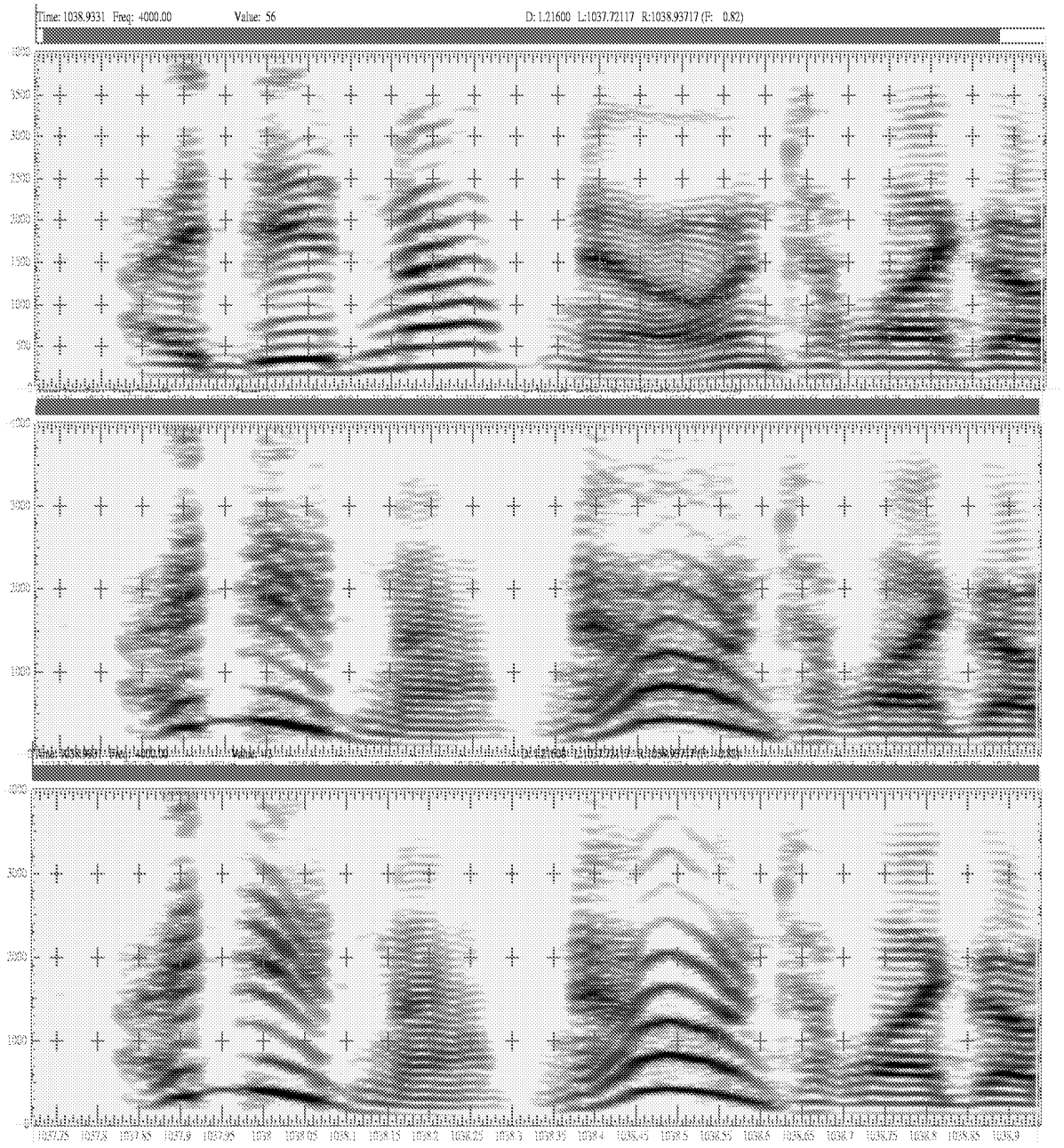

Fig 3. Top, original segment of speech. Center, the same segment, pitch-modified using TD-PSOLA. Bottom, same segment of speech pitch-modified using RTIPS.