SPEECH COMPRESSION USING PITCH SYNCHRONOUS INTERPOLATION

R. Taori, R. J. Sluijter and E. Kathmann

Philips Research Laboratories, Signal Processing Group
Prof. Holstlaan, 4; 5656 AA Eindhoven, The Netherlands
taori@prl.philips.nl

ABSTRACT

This paper presents a new time-domain algorithm for compressing speech signals. Using a novel tool which we will refer to as Time Weighted Average (TWA), a periodically extendable pitch cycle is extracted from the voiced regions in the speech signal. This procedure is carried out every x^{th} pitch period. The discarded $x-1$ pitch periods are recovered using pitch synchronous interpolation (PSI). The computational complexity of the resulting decoder is surprisingly modest and shows reasonable potential of implementation on hardware as primitive as the Intel 8088 μ-processor. Simulation results show that the reconstruction quality is comparable to G.721.

1. INTRODUCTION

For coding speech at low bit rates, significant research is being devoted to exploiting the irrelevancies present in the speech signal. We believe that the answer lies in the pseudo-periodicity exhibited by the speech signal in the "voiced regions". We hypothesise that it should be possible to reconstruct *high* quality speech by transmitting only a representative fraction of the pseudo-periodic data at regular (or varying) intervals. A feasibility study was carried out to validate this hypothesis and in this paper, we attempt to summarize the results.

2. PSI: A NOVEL APPROACH TO SPEECH COMPRESSION

For discussing PSI, we begin with Figure 1. A voiced segment of the speech signal $s[n]$, plotted as the top trace, is marked with dotted lines to indicate the pitch such that n_k defines the location of the k^{th} pitch mark. In the following traces, the speech signal is decomposed into a sequence of short-term overlapping signals $s_k[n]$. The subscript k indicates the k^{th} short-term signal. These short-term signals are obtained simply as a result of multiplying the speech signal by a sequence of

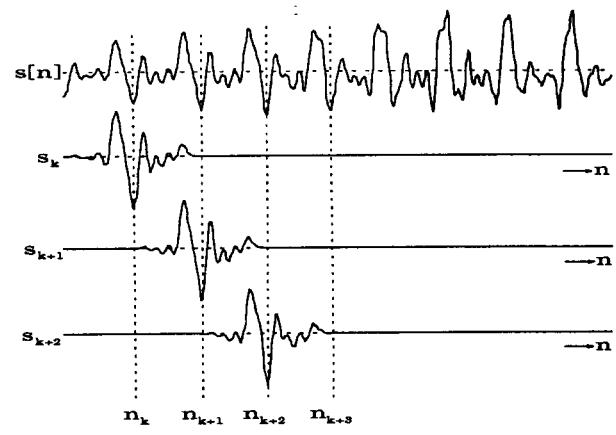

Figure 1: Short-term segments of $s[n]$

analysis windows $w_k[n]$:

$$s_k[n] = s[n] \cdot w_k[n] \tag{1}$$

If these overlapping short-term segments, $s_k[n]$, are added together, we get back the signal $s[n]$. The analysis windows are usually centered around the pitch mark and spans two pitch periods. This procedure in general is called pitch synchronous overlap and add (PSOLA)[1]. Decomposing the signal in this fashion is very useful especially when it comes to altering the prosodic features such as pitch [2]. This ease of manoeuvring the short term signals, combined with the pseudo-periodicity of voiced regions, can be readily exploited for compression purposes. An illustration is exhibited in Figure 2. We retain the short-term signals $s_k[n]$ and $s_{k+3}[n]$, symbolically represented in the Figure by the windows. The short-term signals $s_{k+1}[n]$ and $s_{k+2}[n]$ are discarded. The arrows indicate that the discarded signals are simply replaced by their nearest neighbours. Once again, upon adding all the short-term segments, one can get back a perceptually similar copy of the original signal which is plotted at the bottom. Notice that the amount of short-term signals that one can discard is largely coupled to the underlying sta-

0-7803-2431-5/95 $4.00

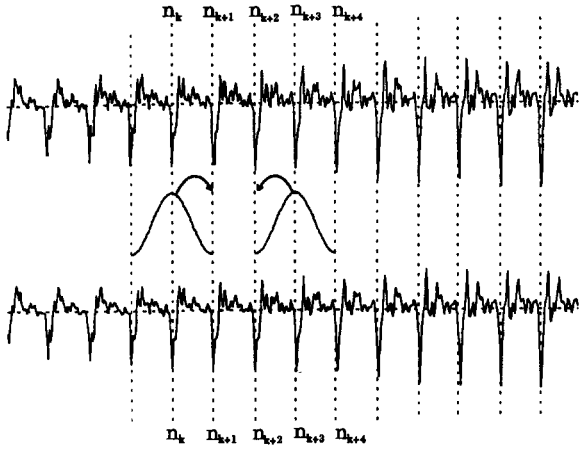

Figure 2: Pitch-period based sample and hold

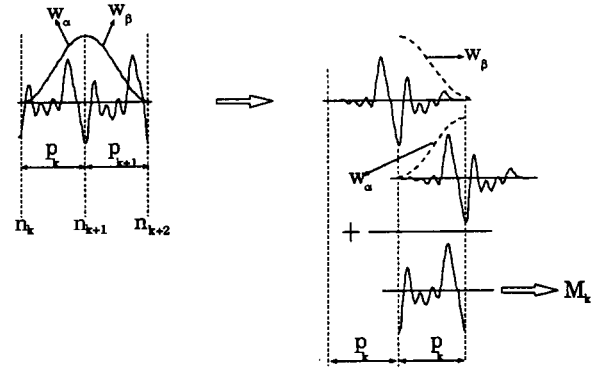

Figure 4: The TWA Mechanism

tionarity of the segment under question. Also notice that, although we retained only $\frac{1}{3}$ of the short-term signals in this illustration, we have effectively retained $\frac{2}{3}$ of the original signal $s[n]$. One can, however, achieve more compression over the same interval by using single pitch periods instead of two pitch periods.

Consider, for example, a decimation $r+1$ scenario depicted in Figure 3. Here we retain the k^{th} pitch period, $M_k[n]$, discard the next r pitch periods, and then the $k+r+1^{st}$ pitch period denoted by $M_{k+r+1}[n]$ is retained. The length of the k^{th} pitch period is denoted by p_k. The retained pitch periods are then coded prior to transmission/storage. The task now is to reconstruct the missing r pitch periods. This is achieved using PSI, which is explained next.

Repeating the pitch periods $M_k[n]$, which have been extracted from the speech signal on an integer grid at 8 KHz, may often give rise to discontinuities. Also, a sample-wise interpolation does not make any sense because the parent pitch-periods M_k and M_{k+r+1}, which would be used to recover the missing r pitch periods, are generally unequal in length. What we need is the capability of making small manoeuvres that we had with the two pitch periods. This, however, cannot be achieved by simply repeating the pitch period $M_k[n]$ because a discontinuity is almost guaranteed to occur in gluing the pitch periods. One could think of incorporating oversampling procedures and so forth, but that makes the system far too complicated.

So, we introduce TWA, a mechanism which circumvents the need to oversample the speech waveform, but at the same time obtain a segment of data which is a pitch period long and is repeatable without any discontinuity. Consider the samples between the pitch marks n_k and n_{k+2}:

$$s[n_k], \ldots, s[n_{k+1}-1], s[n_{k+1}], \ldots, s[n_{k+2}-1]$$

They do not suffer from any discontinuity as we transit from the sample $s[n_{k+1}-1]$ to the sample $s[n_{k+1}]$ as these samples are derived from the original speech signal itself. Our aim is to maintain this natural transition when repeating the pitch period. With this objective in mind, we redefine $M_k[n]$, as being the result of TWA, such that

$$M_k[n] = s[n]w_\alpha[n] + s[n+p_k]w_\beta[n+p_k] \qquad (2)$$

where $w_\alpha[n]$ and $w_\beta[n]$ could, for instance, be the "raised-cosine" function,

$$w_\alpha[n] = \begin{cases} 0.5 - 0.5\cos(R) & , \quad n_k \le n < n_k + p_m \\ 1 & , \quad n_k + p_m \le n < n_k + p_k \\ 0 & , \quad \text{otherwise} \end{cases} \qquad (3)$$

and

$$w_\beta[n] = \begin{cases} 0.5 + 0.5\cos(F) & , \quad n_{k+1} \le n < n_{k+1} + p_m \\ 0 & , \quad \text{otherwise} \end{cases} \qquad (4)$$

where R and F are respectively

$$R = \left(\pi \frac{n - n_k}{p_m - 1}\right), \qquad F = \left(\pi \frac{n - n_{k+1}}{p_m - 1}\right)$$

while

$$p_m = min\{p_k, p_{k+1}\}.$$

The TWA mechanism is explained graphically in Figure 4.

Once $M_k[n]$ is calculated using Equation (2), it is periodically extended to create a twin wavelet, $M'_k[n]$, in the following way:

$$M'_k[n] = \begin{cases} M_k[n] & , \quad n_k \le n < n_k + p_k \\ M_k[n - p_k] & , \quad n_k + p_k \le n < n_k + 2p_k \\ 0 & , \quad \text{otherwise} \end{cases} \qquad (5)$$

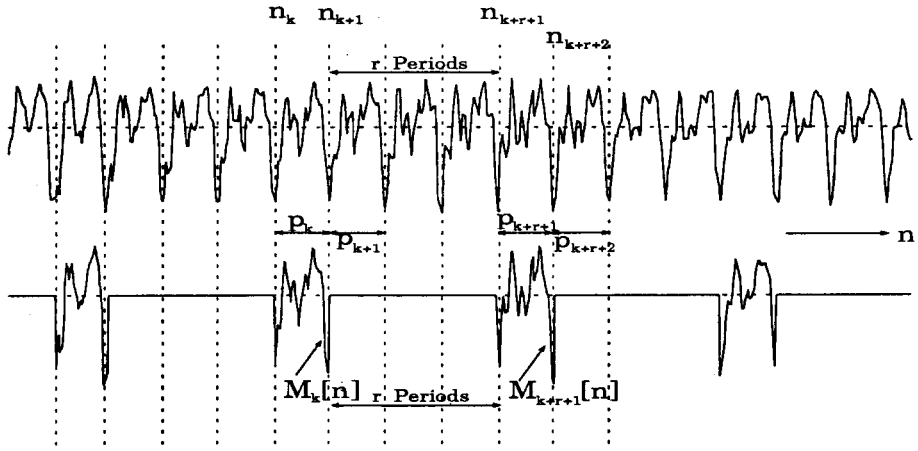

Figure 3: Speech Waveform Decimation

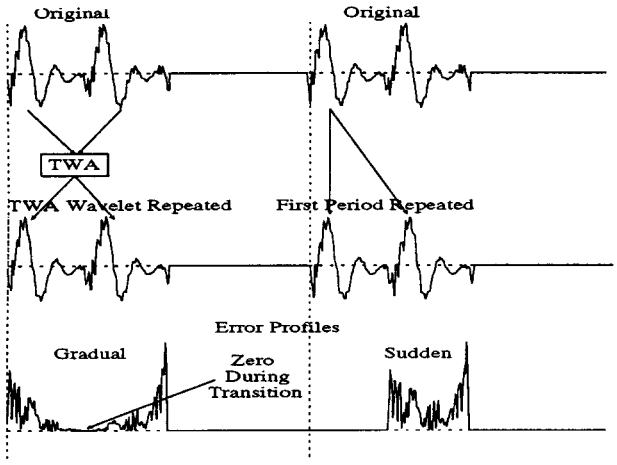

Figure 5: Perceptual Significance of TWA

Now, we see that $M'_k[n_{k+1}-1] = s[n_{k+1}-1]$ and $M'_k[n_{k+1}] = s[n_{k+1}]$, which is exactly how it existed in the original signal.

The perceptual significance of this mechanism is that, when a TWA-wavelet is repeated, there is zero error at the period boundary, when compared to the original speech signal. This is illustrated in Figure 5. For comparison, we have plotted on the right hand half, a simple repitition of the first period. The bottom trace shows the two error profiles. Notice the **gradual** characteristic of the error curve in the TWA-wavelets as compared to the **sudden** jump in case of simple repitition. As will be evident in the discussion to follow, the sharp error peaks at the end points are windowed away during synthesis, thereby leaving a very small error.

Now, without any loss of generality, we home in on a specific case, where the decimation ratio is fixed to equal 4. Therefore the number of discarded wavelets, $r = 3$. In Figure 6, we show two wavelets, $M_0[n]$ and $M_4[n]$, which have been obtained from the speech signal using TWA (Equation (2)). They are periodically extended to obtain $M'_0[n]$ and $M'_4[n]$, using Equation (5), and are plotted in the second trace. Now, the interpolation procedure is applied in the following way: $M'_1[n]$, for instance, is calculated as

$$M'_1[n] = \frac{3}{4}M'_0[n-p_1] + \frac{1}{4}M'_4[n+p_2+p_3] \quad (6)$$

The two parent wavelets, $M'_0[n]$ and $M'_4[n]$ in this case, are first aligned around n_2 by applying appropriate shifts ($M'_0[n]$ is delayed by p_1, while $M'_4[n]$ is advanced by $p_2 + p_3$) prior to performing interpolation. Similarly, $M'_2[n]$ and $M'_3[n]$ are calculated. A reconstructed voiced segment, $\hat{s}_v[n]$ can now be obtained by summing up all the wavelets, $M'_k[n]$ in the following manner:

$$\hat{s}_v[n] = \sum_k M'_k[n]w_k[n] \quad (7)$$

over all k in that voiced segment. The synthesis windows used, should satisfy the following constraint:

$$w_k[n] + w_{k-1}[n] = 1, n_k \leq n \leq n_{k+1}. \quad (8)$$

A possible *synthesis window*, once again, could be a raised cosine-type window function as shown in Figure 6 in dotted lines.

3. RESULTS

A prototype system has been built that operates at around $5kb/s$ on an average. In this system a fixed decimation ratio of 4 was used. PSI was applied only

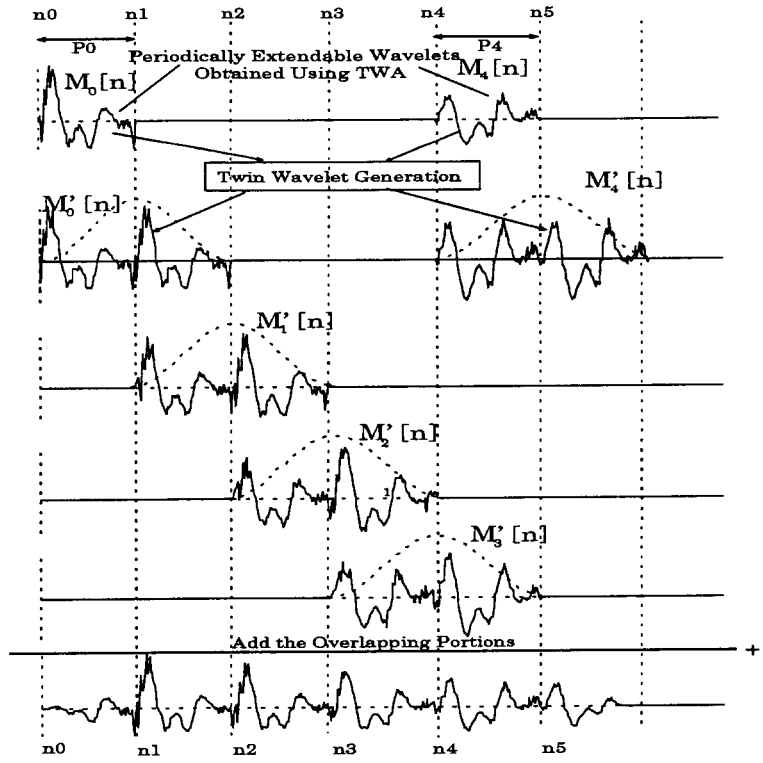

Figure 6: Illustration of PSI for a Fixed Decimation-Ratio (r = 3)

in the voiced regions of the speech segments. The voiced and unvoiced segmentation was performed interactively. The coding of the voiced segments after decimation was peformed using simple differential coding techniques with a fixed second-order predictor and a logarithmic quantizer, thereby giving a bit rate of 6.75 kb/s in the voiced regions. The unvoiced segments can be reconstructed using a sixth-order LPC snthesiser and free running noise generator at a bit-rate of about 1.45 kb/s. The quality of the reconstruction results, approaches the *G*.721 standard.

4. CONCLUSION

We have proposed here, a new time-domain algorithm that efficiently compresses speech signals, by exploiting the irrelevancies present in the acoustic waveform. The efficiency is achieved by using the TWA mechanism, which eliminates the use of oversampling techniques. As a result, the algorithm complexity is much lower when compared to the contemporary proposals such as the PWI coder of Kleijn [3]. The technique should be seen as a two-stage coding scheme, where the speech signal is first decimated on a pitch period basis, and the decimated part can be further coded using existing coding methodology. Depending on the type of coding scheme used, one can build a family of codecs varying in complexity and bit-rates. For instance, a simple DPCM-like coding technique yields a decoder which is extremely simple and can be implemented on extremely inexpensive hardware (eg. Intel 8088 μ-processor). Finally, we would like to point out that for an optimal implementation of this algorithm one must adaptively choose the decimation ratio rather than fixed one, which is used here for illustration.

5. REFERENCES

[1] F. Charpentier and E. Moulines, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using dyphones," in *Eurospeech89*, vol. 2, pp. 13–19, 1989.

[2] W. D. E. Verhelst, "On the quality of speech produced by impulse driven linear systems," in *Proc. ICASSP*, pp. 501–504, 1991.

[3] W. B. Kleijn, "Encoding using prototype waveforms," *IEEE Trans. Speech, Audio Processing*, vol. 1, pp. 386–399, Oct. 1993.