ABSTRACT

There is an increasing demand for the low-cost and high-efficient speech coders used in personal multimedia communications and Voice over IP applications. In this paper the general aspects of G.729 8kb/s CS-ACELP speech coder is presented, then several methods to reduce its complexity with a very slight degradation in speech quality is proposed. The more efficient proposed methods for encoder are “fast open-loop pitch estimation based on pre-search in down sampled signal domain” and “using previously computed pitch delay instead of current pitch based on the difference between the previous and the current LSP coefficients”, and for decoder is “low-cost postfiltering by using an efficient long term postfilter”. The performance of the proposed methods was measured in terms of segmental SNR and DMOS listening test.

1. INTRODUCTION

Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP) coding scheme used in ITU-T G.729 [1] is a CELP-based coding scheme [2]. CELP coding that was first proposed in 1984 with the goal of achieving high speech quality for low bit-rates is an effective scheme for low or medium bit-rates (4 kb/s to 16 kb/s) speech coding but it initially had a very high computational complexity. CELP-based speech coders use the Linear Prediction approach with generation of excitation signal by codebook structure and are realized in the Analysis-by-Synthesis (AbS) optimization [3].

Since 1984, there have been many efforts on complexity reduction methods for CELP coders [4]. Three of the best techniques were standardized by ITU-T in G.728 16 kb/s low-delay CELP (LD-CELP) [5], G.729 8 kb/s CS-ACELP and G.723.1 5.3 kb/s ACELP [6].

The CS-ACELP coding scheme was originally (1996) designed for wireless applications, but is applicable to multimedia communications as well. Annex A of G.729 [7] is a reduced complexity version of the CS-ACELP speech coder at the expense of a slight degradation in speech quality and was designed explicitly for simultaneous voice and data applications that are prevalent in low bit-rate multimedia communications.

These speech coders can be used in many applications such as IP-telephony and videoconferencing. These applications, however, require the low complexity coders since they are cost sensitive due to fierce price competition. Besides, in many applications, a large number of voice channels have to be fit into a given system chassis such as a DSP.

In this paper we propose several methods to reduce the computational complexity of G.729 CS-ACELP speech coder and the complexity of G.729A, too. But the main goal is computational saving without perceptible degradation in speech quality. In the following, we first describe the CS-ACELP algorithm in section 2, and then present the methods to reduce the computational complexity of the CS-ACELP in section 3 that were written in ANSI C codes and then present the simulations and listening tests results in section 4. Finally the conclusion is addressed in section 5.
2. THE CS-ACELP ALGORITHM

Figure 1 shows the block diagram of the CS-ACELP encoder. The coder processes input signals on a frame-by-frame and subframe-by-subframe basis. The frame length is 10 ms and consists of two 5 ms subframes. The first important operation is LP analysis that is done to estimate the spectral envelope characteristics. In this way, the speech signal is expressed in terms of the computed linear prediction coefficients. The quantized LP coefficients are used in the synthesis filter of this coder. The excitation of this filter consists of two parts: The first one is an adaptive codebook vector to represent the periodic component in the excitation signal (pitch structure of the voiced sounds); and the other is a fixed-codebook vector that represents unvoiced sounds. These two codebook vectors scaled by the respective gains are added to construct the excitation of the synthesis filter. The synthesized speech is constructed in such a way that the minimum distortion relative to the original speech is produced while having the best searched codebook vectors.

In decoder first, the parameters are extracted from the received bit-stream. Then, the excitation is constructed by adding the adaptive and fixed-codebook vectors scaled by their respective gains. The speech signal is constructed by filtering the excitation through the LP synthesis filter. The reconstructed speech is passed through the post-filters to enhance the perceptual quality. Finally, the post-processing is performed on the reconstructed speech.

3. METHODS TO REDUCE THE COMPLEXITY OF THE CS-ACELP ALGORITHM

In this section we propose several methods to reduce the complexity of G.729 and G.729 annex A codecs. These methods are also applicable to other annexes such as D and E.

3.1. Fast open-loop pitch estimation

According to ITU-T C codes, relatively a high percentage of time required for encoding, is consumed for open-loop pitch analysis. Therefore faster estimation of the open-loop pitch period can be of interest.

3.1.1 The open-loop pitch estimation, in the CS-ACELP algorithm, is done as follows:

\[ R(k) = \sum_{n=0}^{79} s_w(n)s_w(n-k) \]  

are found in the following three ranges:

\[ i = 1: k = 20,\ldots,39 \]
\[ i = 2: k = 40,\ldots,79 \]
\[ i = 3: k = 80,\ldots,143 \]

where \( s_w \) is the output of the perceptual weighting filter. Then the retained maxima \( R(t_i) \), \( i = 1,2,3 \) are normalized through:

\[ R'(t_i) = \frac{R(t_i)}{\sum_{n=0}^{79} s_w^2(n-t_i)} \]  

where \( t_i \) is the selected delay for the \( i \)th range. The open-loop pitch (\( T_{op} \)) is selected among the three normalized correlations by favoring the smaller delays. In our method \( s_w \) is 4:1 decimated in the three ranges and furthermore \( k \) is down sampled by a factor of 2 in the third range. The 4:1 decimation technique is like a pre-search, which quickly zooms on the most likely region of the pitch period. In this way the computations needed for (1) and (2) are less than \( \frac{1}{4} \) of the computations needed for the
conventional full search. After computing $T_{op}$, the full search is done in the closed-loop pitch analysis around $T_{op}$ to find the pitch period more accurately. Informal listening test shows that this complexity saving leads to perceptually intangible degradation in voice quality. In G.729A standard, (3) has been used instead of (1) and furthermore $k$ has been down sampled by a factor of 2 in the third range. Thus the computational saving in G.729A for 4:1 decimation is less than that of G.729.

$$R(k) = \sum_{n=0}^{29} s_w(n)s_w(2n-k)$$ (3)

The Segmental Signal–to-Noise Ratio (SEGSNR) and the percentage of Computational Reduction (C. R.) in open-loop pitch analysis for different decimations have been summarized in Table 1 for G.729 and G.729A, respectively. It must be noted that these numbers are averaged over male and female speakers.

3.1.2 The second method to reduce the complexity of open-loop pitch analysis is using the adaptive codebook delay of the previous subframe, instead of computing $T_{op}$ for current frame. This method is enabled by three important properties of the pitch [8], which are as follows: a) the pitch contour begins at the unvoice-voice boundary or silence-voice boundary, b) the pitch contour is a smooth curve, c) the pitch contour exists only in the voiced speech. Thus in a continuous speech, the pitch search space has a small range in the voiced speech. Therefore if the difference between the waveforms of two consecutive frames is larger than a threshold $\lambda$, the current $T_{op}$ is estimated according to 3.1.1, otherwise $T_{op}$ is replaced by the adaptive codebook delay of the previous sub-frame. The process is as follows:

$$\text{Diff}_i = \sum_{k=1}^{10} [\text{LSP}_i(k) - \text{LSP}_{i-1}(k)]^2$$ (4)

if \( \text{Diff}_i > \lambda \) then compute $T_{op}$
else $T_{op} = \text{[adaptive codebook delay]}_{i-1}$

where $\text{LSP}_i(k)$ is the $k$th Line Spectral Pair coefficient at the $i$th frame. It must be noted that this method is useful for simulation on PCs, but for DSP implementation is not very effective. The simulation results for open-loop pitch analysis with different threshold $\lambda$ have been listed in Table 2.

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>SEGSNR (dB)</th>
<th>C. R. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Male</td>
<td>Female</td>
</tr>
<tr>
<td>0.0</td>
<td>10.05</td>
<td>10.79</td>
</tr>
<tr>
<td>0.005</td>
<td>9.96</td>
<td>10.74</td>
</tr>
<tr>
<td>0.008</td>
<td>9.88</td>
<td>10.71</td>
</tr>
<tr>
<td>0.010</td>
<td>9.84</td>
<td>10.66</td>
</tr>
<tr>
<td>0.0125</td>
<td>9.77</td>
<td>10.62</td>
</tr>
</tbody>
</table>

Table 2. Simulation results for different $\lambda$ in the modified G.729.

3.2. Efficient long-term post-filtering

A significant portion of the computational complexity in a CS-ACELP decoder is due to the post-filtering, which includes a long-term post-filter, a short-term post-filter and a tilt compensation filter. According to the CS-ACELP algorithm, the long-term post-filter is defined by (5) where $T$ is the pitch delay, $g_l$ is the gain coefficient and $\gamma_p$ controls the amount of long-term post-filtering.

$$H_p(z) = \frac{1}{(1 + \gamma_p g_l z^{-T})}$$ (5)

$T$ and $g_l$ are extracted from the reconstructed speech signals. Their computation is similar to that of the pitch delay and gain in closed-loop pitch analysis in encoder, and thus is time consuming. To achieve a faster decoding, we use a more efficient long-term post-filter. In this approach the pitch delay from the adaptive codebook index is used rather than recompute the pitch delay from the reconstructed signals. In this way we can achieve up to 55% and 18% complexity reduction in the post-filtering for G.729 and G.729A respectively.
3.3. Fast adaptive codebook search and gain estimation

The closed-loop pitch is found in the adaptive codebook search. This operation and gain estimation need a lot of encoder computations. The closed-loop pitch delay is the index $k$ (a delay around the open-loop pitch $T_{op}$) that maximizes the term:

$$R(k) = \frac{\sum_{n=0}^{39} x(n)y_k(n)}{\sqrt{\sum_{n=0}^{39} y_k^2(n)}}$$

(6)

where $x(n)$ is the target signal and $y_k(n)$ is the past excitation at delay $k$ convolved with the impulse response of the perceptually weighted synthesis filter $h(n)$. After determining the closed-loop pitch delay, the adaptive codebook gain $g_p$ is computed as follows:

$$g_p = \frac{\sum_{n=0}^{39} x(n)y(n)}{\sum_{n=0}^{39} y^2(n)}$$

(7)

where $y(n)$ is the $h(n)$ convolved with the adaptive codebook vector $v(n)$ obtained from the adaptive codebook search. Experiments show that over 95% of $h^2(n)$ energy is concentrated in the first five pulses. In our method $h(n)$ is modified to reduce the computations needed for the adaptive codebook search and gain estimation as follows:

$$h(k) = 0 \quad ; 10 \leq k \leq 39$$

Leaving 10 first samples of $h(k)$, instead of 5, unmodified is to obtain a smaller degradation in voice quality. In this way, we in average get about 14% complexity reduction in the adaptive codebook search and gain estimation.

Another way to reduce the required computations is to down sample both $x(n)$ and $y_k(n)$ in (6). For example down sampling by a factor of 2 can save 6% of the needed computations. This change does not lead to an audible degradation in voice quality for G.729, but the last two proposed ways do not lead to a favorite complexity reduction or speech quality for G.729A.

4. PERFORMANCE EVALUATION

4.1. Test methodology

We have done several subjective qualification tests by the Degradation Category Rating (DCR) test [9]. In DCR test method, the listeners first hear the original speech, afterwards its processed version. Then they judge the quality degradation in the second speech compared to the preceding reference speech. Each degradation judgment is done on a 5-point Degradation Mean Opinion Score (DMOS). Score 5 means that the degradation is not perceptible (inaudible), score 4 means it is perceptible but not annoying and scores 3, 2 and 1 are rated by the listeners when the degradation is slightly annoying, annoying and very annoying, respectively.

We prepared the speech data, which consisted of ten different sample speeches spoken by five males and five females in English where each sample was 5 sec long. Ten listeners participated in the experiments and they first heard the decoded speech proposed by the CS-ACELP as the reference speech, then the modified versions of the CS-ACELP, each for several times. The DMOS of each experiment was collected and averaged over all listeners.

In addition to subjective tests we applied one of the objective speech measurements, the segmental SNR, to the variety of the speech data.

4.2. Simulations and tests results

Table 3 illustrates the segmental SNR for two different combinations of the proposed methods, in comparison with original G.729 and G.729A standards. Method (1) is representative of the methods proposed in 3.1 and 3.2 combined together, with parameter $\lambda$ equal to 0.005. Method (2) is the same as (1), but for $\lambda$ equal to 0.008. Computational savings made by methods (1) and (2) for open-loop pitch estimation in G.729 encoder are 83% and 87%, respectively, while an equal saving of 55% has been achieved by both methods for post-filtering in G.729 decoder. Furthermore computational savings obtained by methods (1) and (2) for open-loop pitch estimation in G.729A encoder are 57% and 65%, respectively, while an equal saving of 18%
has been achieved by both methods for post-filtering in G.729A decoder.

Table 4 illustrates the DCR test results of four different combinations of the proposed methods compared to G.729 standard, under noise-free condition. The methods (1) and (2) are the ones used in Table 3. Method (3) is representative of the methods proposed in 3.1, 3.2 and 3.3 combined together, with parameter $\lambda$ equal to 0.005. Method (4) is the same as (3) but for $\lambda$ equal to 0.008. Computational savings obtained by methods (3) and (4) are the same as that of methods (1) and (2) respectively, but with 20% complexity reduction in the adaptive codebook search and gain estimation for encoder, as well. Table 5 shows the DCR test results for methods (1) and (2) compared to G.729A standard, under noise-free condition.

As shown in Tables 3-5, the performance of the methods (1) and (2) is almost equivalent to that of the original CS-ACELP. Furthermore, the quality of the methods (3) and (4) degrades slightly compared to that of the original CS-ACELP, yet it is close to the performance of the original G.729 standard.

5. CONCLUSION

In this paper we have discussed several techniques to reduce the computational complexity of 8 kb/s CS-ACELP speech coder, for applications that require the low complexity coders such as IP-telephony.

These techniques have been applied to open-loop pitch analysis, the adaptive codebook search and gain estimation, and the long-term post-filtering. As detailed in 3.1, 3.2 and 3.3, we have got to a high computational reduction especially in open-loop pitch analysis, with a very slight degradation in the segmental SNR. Also, listening tests have shown that the proposed methods have achieved the speech quality very close to that of the CS-ACELP speech coder under noise-free condition, for both G.729 and G.729A standards.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Male</th>
<th>Female</th>
<th>Ave.</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.729</td>
<td>4.84</td>
<td>4.91</td>
<td>4.88</td>
</tr>
<tr>
<td>G.729A</td>
<td>4.41</td>
<td>4.74</td>
<td>4.58</td>
</tr>
</tbody>
</table>

Table 4. DCR test results of different methods compared to G.729 standard.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Male</th>
<th>Female</th>
<th>Ave.</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.729</td>
<td>4.60</td>
<td>4.38</td>
<td>4.49</td>
</tr>
<tr>
<td>G.729A</td>
<td>4.23</td>
<td>4.33</td>
<td>4.28</td>
</tr>
</tbody>
</table>

Table 5. DCR test results of different methods compared to G.729A standard.

6. REFERENCE


