239

PAPER Efficient Candidate Scheme for Fast Codebook Search in G.723.1

Rong-San LIN^{†a)}, Member and Jia-Yu WANG[†], Nonmember

SUMMARY In multimedia communication, due to the limited computational capability of the personal information machine, a coder with low computational complexity is needed to integrate services from several media sources. This paper presents two efficient candidate schemes to simplify the most computationally demanding operation, the excitation codebook search procedure. For fast adaptive codebook search, we propose an algorithm that uses residual signals to predict the candidate gain-vectors of the adaptive codebook. For the fixed codebook, we propose a fast search algorithm using an energy function to predict the candidate pulses, and we redesign the codebook structure to twin multi-track positions architecture. Overall simulation results indicate that the average perceptual evaluation of speech quality (PESQ) score is degraded slightly, by 0.049, and our proposed methods can reduce total computational complexity by about 67% relative to the original G.723.1 encoder computation load, and with perceptually negligible degradation. Objective and subjective evaluations verify that the more efficient candidate schemes we propose can provide speech quality comparable to that using the original coder approach.

key words: G.723.1, fast codebook search, speech, low computation, twin multi-track

1. Introduction

The ITU-T G.723.1, G.729.1 and G.729 [1]-[3] speech coders are considered the best standards for very low bit rate telephony services, and they have been suggested for use in H.323 Internet phone systems and the H.324 digital videophone service in public switching telephone network (PSTN) systems [4], [5]. The G.723.1 speech coder, proposed by the International Telecommunication Union (ITU), has been used on the Internet extensively, such as the builtin applied software "NetMeeting" in Microsoft's windows operation system and Voice over IP (VOIP) communication systems [6], [7]. Additionally, multimedia communication is integrated into a personal information machine (PIM) nowadays [8], and due to the latter's limited computational capability, the need arises for low computational complexity coders to match different working platforms and integrate services of media sources. For an Internet or wireless speech communicator, heavy computation uses more power and contributes to higher pricing of the communicator or reduced battery life. If we can lower the computational needs of the speech coder for multimedia communication, the PIM may have enough computational capacity and power to han-

Manuscript revised September 10, 2011.

dle other media tasks. Therefore, reduction of computational complexity for the speech coder is desirable for modern communication systems.

The G.723.1 coder is based on principles of linear prediction analysis-by-synthesis (AbS) coding, and attempts to minimize a perceptually weighted error signal. A coder structure of this type can achieve high voice quality and low bit rate. However, a shortcoming of this technology is that the encoder requires much computational complexity to search the codebook and perform the estimated gainvector calculation. It is due to the AbS scheme, used to efficiently model the excitation signal, that the adaptive codebook (ACB) and fixed codebook (FCB) search are the G.723.1's most computationally demanding functional routines. Since G.723.1 uses a fifth-order pitch predictor to model the periodic component of the excitation signal, and pitch delay and gains are searched simultaneously, the computational complexity generated by this procedure results in a heavy computation load. For the FCB stochastic code excited signal, the G.723.1 adopts the multi-pulse maximum likelihood quantization (MP-MLQ) and the algebraic code excited LPC (ACELP) for high rate (6.3 kbit/s) and low rate (5.3 kbit/s) coders, respectively. Lee and Park et al., [9] analyze the distribution of computational load for the encoding process of the G.723.1 with Samsung's DSP chip in a cost-effective implementation. As Table 1 shows, the MP-MLQ and ACB codebook search procedures constitute over 55% and 23%, respectively of the computations required in the G.723.1 encoding process. To simplify these computations, a good trade-off between speech quality and codebook search complexity is required.

Several efficient fast search methods have been proposed in the existing literature to reduce codebook search complexity [10]–[17]. For the ACB search approach, Jung et al., [14], [15] proposed a first-order closed-loop pitch predictor to predict pitch gains and pitch lag of the fifth-

 Table 1
 The distribution of CPU computational load in the encoders.

Function description	G.723.1, 6.3 kbps	G.723.1, 5.3 kbps
LPC	1.4%	1.6%
Open-loop pitch	5%	5.8%
LSP	7.3%	8.4%
Filtering	8.6%	10%
ACB	23.2%	26.8%
FCB	54.5%	47.4%

Manuscript received June 13, 2011.

[†]The authors are with the Department of Computer Science and Information Engineering Southern Taiwan University, No. 1, Nan-Tai Street, Yongkang Dist., Tainan City 71005, Taiwan.

a) E-mail: rslin@mail.stut.edu.tw

DOI: 10.1587/transinf.E95.D.239

order closed-loop pitch predictor. Considering FCB search, Jung's proposed approach used a depth-first search instead of original focused search. Additionally, to reduce search computation, they used the ACELP structure instead of the original MP-MLQ structure in the G.723.1 (6.3 kbps) coder. Jung's proposed codebook structure arrangement could lose significant excitation pulse at the last track. Lin et al., [16] provided a search method to reduce the computation required of MP-MLQ and Chen et al., [17] provided two complexity scalability approaches to reduce the computation requirements of the ACELP and MP-MLQ.

In this paper, we propose two efficient candidate schemes for fast search of the ACB and FCB, respectively. First, we propose a fast ACB search algorithm to reduce computational complexity. This scheme utilizes third- order open-loop pitch gains and first-order closed-loop pitch predictor to predict the pitch lag and candidate gain-vectors of the ACB. As for the MP-MLQ search, to further improve speech quality of [15], [16] we proposed twin multi-track positions architecture for even and odd subframes, respectively.

This paper is organized as follows: in Sect. 2, the G.723.1 speech coding algorithms are briefly reviewed; in Sect. 3, we describe the proposed fast codebook search methods for G.723.1 coder. To verify the efficiency of the proposed techniques, experimental results are presented in Sect. 4. Conclusions are presented in Sect. 5.

2. ITU-T G.723.1 Speech Coder

The ITU-T G.723.1 encoder operates on blocks (30 ms frames) of 240 samples each. Each frame is first divided into four subframes of 60 samples each. For every subframe, a set of 10th order linear prediction coefficients (LPC) are computed using the unprocessed input signal. The LPC set of the last subframe is converted to LSP parameters and is then quantized using a predictive split vector quantization (PSVO) and transmitted to the decoder. The un-quantized LPC are used to construct the short-term perceptual weighting filter, which is used to filter the entire frame and to obtain the perceptually weighted speech signal. For every two subframes, the open-loop pitch period is computed using the weighted speech signal. This pitch estimation is searched in the range from 18 to 142 samples. Every subframe speech signal is then encoded by the ACB and FCB search procedures. ACB search is performed using a fifthorder pitch predictor. Finally, stochastic excitation pulses are approximated by the MP-MLQ excitation for high bit rate (6.3 kbit/s), and an ACELP for low bit rate (5.3 kbit/s).

2.1 Standardized ACB Search

The ACB search, in G.723.1 estimates the closed-loop pitch lag and gains simultaneously. The ACB is searched by minimizing the square error (*MSE*) between the weighted speech signal, t(n), and the weighted synthesis speech of the pitch predictor, $\hat{p}(t)$, and this is denoted as:

$$MSE_{ACB} = \sum_{n=0}^{N-1} \left(t(n) - \sum_{i=-2}^{2} \beta_{ki} \hat{p}(n-L+i) \right)^2$$
(1)

For a given pitch lag *L*, the optimum pitch gains β_{ki} are obtained by MSE_{ACB} as stated in Eq. (1), where β_{ki} values are the pitch predictor gains, *N* is the subframe length, and *k* are the adaptive gain-codebook indices. In practice, the optimum pitch lag and gains are searched to maximize the following term in (2).

$$\beta_{opt} = \min_{\beta} \left\{ \sum_{n=0}^{N-1} t^{2}(n) - \Phi(\beta) \right\}, \text{ where} \\ \Phi(\beta) \\ = \sum_{n=0}^{N-1} \left\{ 2t(n) \sum_{i=-2}^{2} \beta_{ki} \hat{p}(n-L+i) - \left(\sum_{i=-2}^{2} \beta_{ki} \hat{p}(n-L+i) \right)^{2} \right\}$$
(2)

The closed-loop pitch lag is computed as a small differential value around the open-loop pitch lag estimate or the previous subframe pitch lag. In other words, for subframes 0 and 2 the closed loop pitch lag is selected from around the appropriate open-loop pitch lag in the range ± 1 and coded using 7 bits. For subframes 1 and 3, the closed-loop pitch lag is coded differentially using 2 bits, and may differ from the previous subframe lag only by -1, 0, 1, or 2. Therefore, every subframe on average requires 3.5 iterations to search the gain-codebook. The quantized and decoded pitch lag values will be referred to as L_i from this point on. The pitch predictor gains are vector quantized using two gain-codebooks (GB) with 85 or 170 entries for the high bit rate and 170 entries for the low bit rate. The 170 entry codebook is the same for both rates. For the high rate if L_0 is less than 58 for subframes 0 and 1 or if L_2 is less than 58 for subframes 2 and 3, then the 85 entry codebook is used for the pitch gain quantization. Otherwise the pitch gain is quantized using the 170 entry codebook. For the 170 entry codebook, Jung et al., [14] analyzed the standard ACB search method used in the G.723.1 coder and found it requires 49415 multiplication operations for every subframe. The contribution of the pitch predictor is then subtracted from the weighted speech signal, t(n) to obtain the weighted residual (speech) signal, r[n], for FCB search.

2.2 Standardized MP-MLQ Codebook Search

After the short-term analysis and long-term prediction, the weighted residual (speech) signal, r[n], is obtained as a new target signal for stochastic excitation processing. The stochastic excitation search, which performs estimation and quantization for the target vector, involves the determination of pulse positions and amplitudes. The criterion of codebook search is to minimize the square error *E* between the target signal, r[n], and the synthesis residual (speech), r'[n], as follows:

$$E = \sum_{n=0}^{N-1} |r[n] - r'[n]|^2$$
(3)

To achieve a good approximation of the target vector, r[n], the encoding process by r'[n] is denoted as:

$$r'[n] = \sum_{j=0}^{n} h[j] \cdot v[n-j], 0 \le n \le N-1$$
(4)

where *N* is the subframe length and v[n] denotes the excitation to the synthesis filter h[n]. v[n] can be expressed as [1]:

$$v[n] = G \sum_{k=0}^{M-1} \alpha_k \delta[n - m_k], 0 \le n \le N - 1$$
 (5)

where $\delta[n]$ is a Dirac function and *G* is the common gain factor for each pulse, m_k denotes the excitation pulse position with $\alpha_k = \pm 1$ for $k = 0, 1, \dots, M - 1$, and *M* represents the number of pulses, which is 6 for even subframes and 5 for odd subframes. There is a restriction on pulse positions, i.e. the positions can either be all odd or all even, which is indicated by a grid bit. Consequently, the purpose of optimization is to estimate the unknown parameters *G*, { α_k } and { m_k } for $k = 0, 1 \dots M - 1$, such that the mean square of the error signal, err[n], is minimized, and is expressed by

$$err[n] = r[n] - r'[n] = r[n] - G \sum_{k=0}^{M-1} \alpha_k h[n - m_k]$$
 (6)

According to the property of maximum likelihood, the cross-correlation function, d[j], between the impulse response, h[n], and the target signal, r[n], is first computed:

$$d[j] = \sum_{n=j}^{N-1} r[n] \cdot h[n-j], 0 \le j \le N-1$$
(7)

Moreover, the optimal gain G_{max} is estimated by:

$$G_{\max} = \frac{\max\{|d[j]|\}_{j=0...N-1}}{\sum_{n=0}^{N-1} h[n] \cdot h[n]}$$
(8)

Finally, the combination of the quantized parameters that yield the minimum mean square of the error signal, err[n], is selected.

In the G.723.1, the optimal combination of pulse positions and gain is encoded. $2 \times C_M^{30}$, combinatorial numbers are used for the pulse positions where M = 5 or 6. However, for real-time applications, the number of combinations of all possible pulse positions is too large to be searched. Thus, reducing the number of combinations of possible pulse positions in the G.723.1 MP-MLQ search algorithm helps improve encoder efficiency.

3. The Proposed Fast G.723.1 Search

In this paper, two candidate schemes for the ACB and the

MP-MLQ search algorithms are proposed to significantly reduce computational complexity. To evaluate the performance of the proposed methods objectively we use method such as PESQ (ITU-T Rec.P862) [18], and subjective evaluation using informal MOS testing.

3.1 Fast ACB Search

The pitch predictor gains are vector quantized and every gain-vector has 20 elements. We need to find the best gain-vector to maximize $\Phi(\beta)$ in Eq. (2) by substituting all gain-vectors of *GB*. For the fifth-order predictor, each gain-vector requires 20*N* multiplications to compute $\Phi(\beta)$ by looking up the codebook gain-table.

For example, considering 170 entries *GB*, to obtain optimal closed-loop pitch lag and the related gain-vector, one must search 4×170 gain-vectors of the ACB for subframes 1 and 3, and 3×170 gain-vectors for subframes 0 and 2. Thus ACB search in G.723.1 requires heavy computation. The efficiency of the G.723.1 encoding speech is improved by reducing the number of gain-vectors searched. We proposed a fast ACB search algorithm for this purpose. Our scheme utilizes third-order open-loop pitch predictor to preselect candidate gain-vectors and the related open-loop pitch lag, followed by the use of the first-order closed-loop pitch predictor to estimate pitch lag.

The flow chart of the proposed scheme is shown in Fig. 1. First the target signal t(n) was filtered by a 1/H(z) filter to generate the excitation signal E[n], and using open-loop pitch lag L and ACB gains to generate the excitation signal $e'_k[n]$. The functions are given by:

$$E[n] = t(n) - \sum_{i=0}^{9} a(i)t(n-i), 0 \le n \le N - 1$$
(9)

$$e'_{k}[n] = \sum_{i=-1}^{i=1} \beta_{ki} e[n-L+i], 0 \le n \le N-1$$
(10)

where a(i) are LPC coefficients, e[n] is the previous excitation signal, and β_{ki} are ACB gain-vectors. The third-order open- loop pitch gain-vectors are searched by using Eq. (11)

$$MSE_{k} = \sum_{n=0}^{N-1} (E[n] - e'_{k}[n])^{2},$$

$$0 < n < N - 1, 0 < k < GB - 1$$
(11)

We adopt Minimum squared error (MSE_k) as the criterion to estimate M candidate gain-vectors from GB (170 or 85 entries) and open-loop pitch lag L_{olp}^3 , where k denotes the indices of GB. The above process is an open-loop search, which requires less computation than a closed-loop search. So each subframe requires 9(GB + N) multiplications by looking up the codebook gain-table. Additionally, in the estimation above, the pitch lag L_{olp}^3 is selected from a range similar to that in Eq. (2) earlier. In the 3-tap open-loop process stage, the proposed method searches $3.5 \times GB$ gainvectors on average to preselect the M candidate gain-vectors



Fig. 1 Flow chart of the proposed fast ACB search algorithm.

and the prediction pitch lag L_{olp}^3 .

A first-order closed-loop pitch lag is then computed to minimize the mean square error between the weighted speech signal and the weighted synthesis speech, $\hat{p}(n - L)$, given by:

$$MSE_{one} = \sum_{n=0}^{N-1} (t(n) - \beta \hat{p}(n-L))^2$$
(12)

where the weighted synthesis speech requires 10(N-1) multiplications for every subframe. In practice, we need to find the best closed-loop pitch lag L_{clp}^1 , to maximize the following term in Eq. (13) by substituting the given M candidate gains β and the pitch lag L. The pitch lag L_{clp}^1 is searched a range similar to that in Eq. (2) earlier. In the 1-tap closed-loop process stage, the proposed method only searches $3.5 \times M$ candidate gain-vectors on average to predict the pitch lag L_{clp}^1 .

$$MAX_{one} = \beta \sum_{n=0}^{N-1} (t(n)\hat{p}(n-L)) - \frac{1}{2}\beta^2 \sum_{n=0}^{N-1} \hat{p}^2(n-L)$$
(13)

In Eq. (13) the first-order predictor requires 2(N + 1) multiplications for every subframe. These estimations are performed before the original ACB search procedure.

Finally, the G.723.1 ACB coding process only searches M candidate gain-vectors with the predicted pitch lags, L_{cln}^1



Fig.2 Average degradation PESQ with regard to different percentages of preselected *GB*.

and L_{olp}^3 . The predicted pitch lags L_{clp}^1 and L_{olp}^3 , may or may not be equal. When $L_{clp}^1 = L_{olp}^3$ the proposed algorithm uses only one parameter, the predicted pitch lag L_{clp}^1 . Otherwise, the algorithm uses both L_{clp}^1 and L_{olp}^3 . Since block " $(L_{clp}^1 = L_{olp}^3)$ " branches stochastically with equal probability in Fig. 1, the G.723.1 ACB coding process requires 1.5 iterations on average for every subframe.

The pitch gains of the fifth-order pitch predictor are computed as in Eq. (2). We need to find the best gainvector, to maximize $\Phi(\beta)$ by substituting these *M* candidate gain-vectors. Therefore the proposed algorithm only tests $M \times 1.5$ candidate gain-vectors for every subframe. It must be noted that the fast ACB search approach proposed by Jung et al. [14] also required testing $85 \times 2 = 170$ (2 iterations) gain-vectors for every subframe, saving 56.16% more in terms of computational load compared to the G.723.1 ACB approach, and the related PESQ score degrades by 0.01.

We estimate speech quality relative to preselected candidate gain-vectors in our experiment. Figure 2 shows the number of candidate gain-vectors M corresponding to different percentages of preselected GB. We observe that the number of candidate gain-vectors from experimentation produces $GB \times 20\% = M$ gain-vectors and can achieve an optimum in terms of speech quality and computational complexity. For example, 20% of GB with lower value MSE_k using Eq. (11), were preselected as candidate gain-vectors. Using GB = 170 entries, M = 34 gain-vectors are obtained. It should be noted that the G.723.1 ACB coding process only searches $M \times 1.5 = 51$ candidate gain-vectors for every subframe. However, the original G.723.1 ACB search must test $170 \times 3.5 = 595$ (3.5 iterations) gain-vectors for every subframe. Therefore, the proposed fast search algorithm can reduce the computational complexity by about 91.42% $(1 - \frac{3}{7} \times 20\%)$ compared to the G.723.1 ACB search. However, the preprocessing for deciding the candidate gainvectors and pitch lags requires an extra computational load of about 18.7% (for 170 entries, 9231 multiplications). The



Fig. 3 PESQ of the proposed method compared with original ACB full search (average degradation 0.029).

preselected 20% gain-vectors from *GB* are used in the experiment, and results show that the average of the PESQ score is degraded slightly by 0.029, relative to the original G.723.1 search procedure. However, the proposed method can dramatically reduce the computational complexity by about 72.7% with perceptually negligible degradation. The true PESQ values of the proposed method compared with original ACB search is shown in Fig. 3.

3.2 Multi-Track Fast MP-MLQ Search

For the 6.3 kbit/s coder, MP-MLQ excitation signal is used, and the coder is based on the analysis-by-synthesis technology. This type of coder structure can achieve high voice quality and low bit rate, but a shortcoming of this technology is that the encoder requires heavy computation complexity. The MP-MLQ search algorithm calculates the *MSE* of both the odd and even pulse positions for each subframe, respectively, and then the least *MSE* is selected. It is pointed out earlier in Eq. (5)–(8) that the MP-MLQ search algorithm entails high computational complexity.

Previously, we proposed a fast search algorithm to reduce computational complexity of the MP-MLQ search algorithm [16]. In this algorithm, we estimated candidate pulse positions and used a single multi-track structure instead of the original MP-MLQ structure to reduce the number of combinations of possible pulse positions. To further improve speech quality, we modified our previous structure of pulse positions in this paper.

The original ACELP method arranged the structure of the excitation pulse positions; the structure can provide good speech quality and low bit rate. The signal vector, b[n] is used in AMR to search the algebraic codebook. To reduce MP-MLQ search complexity, the proposed method uses the signal vector, b[n] and the structure of the ACELP codebook to preselect candidate pulse positions. Then, the original G.723.1 MP-MLQ search algorithm is processed. In other words, the structure of the ACELP codebook combining signal vector, b[n] is used merely to preselect candidate pulse positions.

First every subframe, the 60 samples are divided into

Table 2Multi-track structure of the even subframes.

	Sign	Pulse positions
t_0	±1	0(1), 12(13), 24(25), 36(37), 48(49)
t_1	±1	2(3), 14(15), 26(27), 38(39), 50(51)
t_2	±1	4(5), 16(17), 28(29), 40(41), 52(53)
<i>t</i> ₃	±1	6(7), 18(19), 30(31), 42(43), 54(55)
t_4	±1	8(9), 20(21), 32(33), 44(45), 56(57)
t_5	±1	10(11), 22(23), 34(35), 46(47), 58(59)

Table 3 Multi-track structure of the odd subframes.

	Sign	Pulse positions
t_0	±1	0(1), 10(11), 20(21), 30(31), 40(41), 50(51)
t_1	±1	2(3), 12(13), 22(23), 32(33), 42(43), 52(53)
<i>t</i> ₂	±1	4(5), 14(15), 24(25), 34(35), 44(45), 54(55)
<i>t</i> ₃	±1	6(7), 16(17), 26(27), 36(37), 46(47), 56(57)
<i>t</i> ₄	±1	8(9), 18(19), 28(29), 38(39), 48(49), 58(59)

a multi-track structure as shown in Table 2 (for even subframes) and Table 3 (for odd subframes). Previous research, [15] has the problem of losing significant excitation pulse at the last track for odd subframes, resulting in a degradation of speech quality. Our proposed twin multi-track position structure, thus, overcomes this issue.

$$b_{j}[m]: m = \begin{cases} [12i+2t+j], 0 \le i \le 4, 0 \le t \le 5, even \ subframes\\ [10i+2t+j], 0 \le i \le 5, 0 \le t \le 4, odd \ subframes\\ j = 0, 1 \quad Pulse \ positions \ of \ the \ Multi-track \end{cases}$$

The flow chart of the proposed scheme is shown in Fig. 4. Firstly, the target signal, r[n], was filtered by the A(z) filter to generate the excitation signal, $res_{LTP}[n]$, for every subframe, where an A(z) filter is defined as:

$$A(z) = 1 - \sum_{i=1}^{10} a[i]z^{-i}$$
(14)

where a[i] are LPC coefficients. The pulse-position likelihood-estimate energy vector, b[n][2], [19] is defined as:

A

$$b[n] = \frac{|res_{LTP}[n]|}{\sqrt{\sum_{i=0}^{N-1} res_{LTP}[i] \cdot res_{LTP}[i]}} + \frac{|d[n]|}{\sqrt{\sum_{i=0}^{N-1} d[i] \cdot d[i]}},$$
$$0 \le n \le N - 1 \tag{15}$$

where *N* is the subframe length and the d[n](7) is crosscorrelation function. Let $b_j[m]$ denote one of the *K* largest values of b[n] and m denote the excitation pulse positions.



Fig.4 Flow chart of the proposed fast search algorithm for the G.723.1MP-MLQ.



Fig. 5 Average degradation PESQ with regard to different number K of pulse positions (original MP-MLQ, average PESQ = 3.575).

Firstly, the process preselects the K largest values of b[n] as the candidate pulse positions for odd positions (j = 1) in each track, and likewise for even positions (i = 0). The estimate candidate pulse positions process was performed before the original standard MP-MLQ search procedure. We analyze the average degradation of PESQ relative to the number of candidate pulse positions for every track in the experiment and the results are shown in Fig. 5. It is observed that preselecting K = 3 candidate pulse positions from both the odd and even positions in each track, respectively, can achieve an optimum in terms of speech quality and computational complexity. In this case the number of combinations of pulse positions will be reduced from 1187550 $(2 \times C_6^{30})$ to 37128 $(2 \times C_6^{6\times 3})$ for every even subframe. For odd subframes, the number of combinations of the positions will be reduced from 285012 to 6006. The computational complexity of MP-MLQ coding is therefore significantly reduced by using our proposed method. We preselected 6 candidate pulse positions (K = 3) for every track in the experiment, and results show that the average degradation of



Fig. 6 PESQ of the original MP-MLQ approach in comparison with the original ACELP method and the proposed method.

the PESQ score is about 0.016 relative to the original MP-MLQ search procedure. However, the proposed method reduces the computational complexity by about 92.1% with perceptually negligible degradation. Experimental results are shown in Fig. 6. Comparing the true PESQ values of the proposed method with the original MP-MLQ search approach and original ACELP method, we found that the average PESQ values of the proposed method and the original ACELP method are 3.559 and 3.43 respectively. It must be noted that the proposed method is based on the MP-MLQ method which is a 6.3 kbit/s coding rate while the ACELP method is a 5.3 kbit/s.

4. Overall Performance Analysis

In this paper, we proposed two efficient fast search algorithms for the G.723.1 speech coder. To evaluate the overall performance of the proposed schemes and the original G.723.1 method, subjective preference tests are performed together with objective speech quality evaluation and computational complexity analysis. Subjective speech quality is evaluated via a MOS test, and an objective speech quality measure, PESQ is used.

4.1 Objective Speech Quality and Computational Complexity Evaluation

In our experiments, the fast ACB search and the multi-track fast MP-MLQ search algorithms were simultaneously implemented in the G.723.1 coder. It must be noted that the average PESQ score can be considered as an evaluation of the overall performance, and our experimental results show that the average PESQ score degraded slightly by 0.049, relative to the original G.723.1 coding. The true PESQ values of the proposed methods compared to the original G.723.1 (6.3 kbit/s) approach are shown in Fig. 7.

We propose two fast search algorithms to reduce the computation of the ACB and MP-MLQ coding for the G.723.1 as discussed earlier in Sect. 3.1 and 3.2, respectively. In contrast to the complexity shown in Table 1, we proposed fast search algorithms can reduce the computational complexity by about $23.2\% \times 0.73 \approx 16.9\%$ and $54.5\% \times 0.92 \approx 50.1\%$ relative to the original search compu



Fig.7 PESQ of the original G.723.1 and the integrated two fast search methods.

Methods	Original G.723.1		Proposed	
MOS	Male	Female	Male	Female
1(bad)	0	0	0	0
2	0	0	0	0
3	10	16	15	23
4	154	157	151	148
5(good)	36	27	34	29
Average	4.13	4.06	4.09	4.03

Table 4Results of the listeners' preference test (at 6.3 kbit/s bit rate).

(10 tested speech files x 20 listeners = 200 scores for every column)

tational complexity in the ACB and the MP-MLQ, respectively. Overall, the proposed fast search algorithms can reduce total computational complexity by about 67% relative to the original G.723.1 coding computation load.

4.2 Subjective Speech Quality Evaluation

To verify objective results of the PESQ measurements, a simple and informal mean opinion score (MOS) assessment is also offered in this paper. We implement a subjective quality measurement called the A-B test. Twenty speech files are tested for speech quality evaluation. These speech files were recorded by 10 males and 10 females in a general environment. A total of 20 non-expert participants working in the field of multimedia data compression and processing were invited to perform the test. In the tests, these untrained listeners were asked to give a score from 1 (bad) to 5 (good) based on their preferences, using a headset. These MOS scores are summarized in Table 4, and testing results show that the difference in subjective quality between the proposed method and the original G.723.1 is negligible. Results imply that the listeners cannot distinguish the quality of the original G.723.1 coding from that of the proposed fast search methods. In Table 4, the average MOS score of every column was calculated. For example, $(3 \times 10 + 4 \times 154 + 5 \times 36) \div 200 = 4.13$. To accompany the subjective tests described above, we have made the decoded sound files available at http://faculty.stut.edu.tw/~rslin/IEICElist.htm for subjective evaluation by listening.

5. Conclusions

In this paper, we propose fast ACB search and twin multitrack fast MP-MLQ search algorithms to reduce the computational complexity of the G.723.1 coder. Using the proposed methods, the number of MP-MLQ search pulse positions and ACB search gain-vectors can be reduced. Results of subjective evaluation show that the proposed schemes can produce speech quality equivalent to that of the original G.723.1 coding. Simulation results also show that the average of the PESQ score is degraded slightly, by 0.049, and our proposed methods can reduce total computational complexity by about 67% relative to the original G.723.1 encoding computation load with perceptually negligible degradation.

Acknowledgments

This work was supported by the National Science Council of Republic of China under research contract NSC- 99-2221-E-218 -038. The authors would like to thank the reviewers for their valuable comments and suggestions.

References

- ITU-T Rec. G.723.1, Dual Rate Speech Coder for Multimedia Communications at 5.3 and 6.3 kbit/s, March 1996.
- [2] ITU-T Rec. G.729.1, G.729-based embedded variable bit-rate coder: An 8-32 kbit/s Scalable Wideband Coder Bitstream Interoperable with G.729, May 2006.
- [3] ITU-T Rec.G.729, Coding of Speech at 8 kbit/s using Conjugate Structure Algebraic Code Excited Linear Prediction (CS-ACELP), March 1996.
- [4] ITU-T Rec. H.324, Terminal for Low Bit Rate Multimedia Communication, March 1996.
- [5] ITU-T Rec. H.323, Visual Telephone Systems and Equipment for Local Area Networks Which Provide a Non-guaranteed Quality of Service, March 1996.
- [6] G. Thomsen and Y. Jani, "Internet telephony: Going like crazy," IEEE Spectr., vol.37, no.5, pp.52–58, May 2000.
- [7] D. Minoli and E. Minoli, Delivering Voice over IP Network, John Wiley & Sons, 1998.
- [8] S. Dixit, Y. Guo, and Z. Antoniou, "Resource management and quality of service in third generation wireless networks," IEEE Commun. Mag., vol.39, no.2, pp.125–133, Feb. 2001.
- [9] S.M. Lee, S. Park, and Y. Jang, "Cost-effective Implementation of ITU-T G.723.1 on A DSP Chip," Proc. 1997 IEEE International Symposium on Consumer Electronics, pp.31–34, Dec. 1997.
- [10] J.F. Yang, R.S. Lin, and S. Hu, "Time-varied pitch gain predictor for low bit rate speech coders," IEICE Trans. Inf. & Syst., vol.E85-D, no.4, pp.751–758, April 2002.
- [11] C. Zhang, K. Tang, H. Cui, W. Du, and J. Li, "Efficient pitch predictor algorithm in ITU-T G.723.1," IEEE Conf. Communication Technology Proceedings, vol.02, pp.1727–1729, April 2003.
- [12] Y.C. Chen, R.S. Lin, and F.K. Chen, "A fast search algorithm for adaptive codebook of ITU-T G.723.1 codec," 2007 International

Conference on Advanced Information Technologies (AIT), Session 3-F, p.60, April 2007.

- [13] S.M. Tsai and J.F. Yang, "Efficient algebraic code-excited linear predictive codebook search," IEE Proceedings-Vision, Image, and Signal Processing, vol.153, no.61, pp.761–768, 2006.
- [14] S.K. Jung, K.T. Kim, Y.C. Park, and H.G. Kang, "A fast adaptivecodebook search algorithm for G.723.1 speech coder," IEEE Signal Process. Lett., vol.12, no.1, pp.75–78, Jan. 2005.
- [15] S.K. Jung, Y.C. Park, S.W. Youn, K.T. Kim, and D.H. Youn, "Efficient implementation of ITU-T G.723.1 speech coder for multichannel voice transmission and storage," Eurospeech 2001 – Scandinavia, 2001.
- [16] R.S. Lin, Y.C. Chen, and F.K. Chen, "Low complexity search method for G.723.1 MP-MLQ algorithm," The eighth International Conference on Intelligent Systems Design and Applications, vol.1, pp.158–161, Nov. 2008.
- [17] F.K. Chen, J.F. Yang, and Y.P. Lin, "Complexity scalability for ACELP and MP-MLQ speech coders," IEICE Trans. Inf. & Syst., vol.E85-D, no.1, pp.255–263, Jan. 2002.
- [18] ITU-T Rec. P.862, Perceptual Evaluation of Speech Quality (PESQ), an Objective Method for End-to-end Speech Quality Assessment of Narrowband Telephone Networks and Speech Codecs, Feb. 2001.
- [19] 3GPP TS 26.090: ARM Speech Codec; Speech Transcoding Function, 3GPP, June 2007.



Rong-San Lin was born in Kaohsiung County, Taiwan, in 1958. He received the B.S. and M.S. degrees in electronic engineering from the National Taiwan University of Science and Technology, in 1986 and 1990, respectively, and the Ph.D. degree in Electrical Engineering from the National Cheng Kung University, Taiwan in 2002. Currently, he is an Associate professor in Department of Computer Science and Information Engineering in Southern Taiwan University, Taiwan. His main teaching and research are sig-

nal processing, speech coding, fast algorithm design and DSP application implementation.



Jia-Yu Wang was born in Pingtung, Taiwan, in 1985. He received M.S. degree in Dept. of Computer Science and Information Engineering from the Southern Taiwan University, in 2010. Currently, he is a R&D engineer.