International Journal of Innovative Computing, Information and Control
Volume **8**, Number **7(A)**, July **2012**

ICIC International ©2012 ISSN 1349-4198
pp. **4635–4660**

AN EFFICIENT TRANSCODING SCHEME FOR G.729 AND G.723.1 SPEECH CODECS: INTEROPERABILITY OVER THE INTERNET

Rong-San Lin[1], Jia-Yu Wang[1] and Jeng-Shyang Pan[2]

[1]Department of Computer Science and Information Engineering
Southern Taiwan University
No. 1, Nan-Tai Street, Yungkang Dist., Tainan City 710, Taiwan
rslin@mail.stut.edu.tw

[2]Department of Electronic Engineering
National Kaohsiung University of Applied Sciences
Chien Kung Campus 415, Chien Kung Road, Kaohsiung 807, Taiwan

Received July 2010; revised October 2011

Abstract. *This paper proposes an efficient conversion algorithm for G.729 and G.723.1 speech codecs to reduce computational complexity of the communications between the G.729 and G.723.1 speech codecs. The proposed transcoding method incorporates four processes: line spectral pair (LSP) interpolation, pitch conversion, fast adaptive-codebook search, and fast fixed-codebook search. To reduce search computations, we propose a fast adaptive codebook search 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 that uses an energy function to predict the candidate pulse positions. Other codec parameters are directly converted in parametric levels without executing the decoding process. Simulation results show that the proposed methods can reduce total computational complexity by 65.8%, with a shorter coding delay compared with the commonly used decode-then-encode tandem approach. Objective and subjective evaluations were used to verify that the proposed transcoding scheme provides speech quality comparable to the tandem approach.*

Keywords: Speech coding, G.723.1, G.729, Transcoding, Tandem, Fast codebook search, LSP, Pitch

1. **Introduction.** Speech transmission is the dominant service in telecommunications networks and in the multimedia domain, specifically the emerging Voice over IP (VoIP) protocol. VoIP is based on existing data network services [1,2] and speech processes were proposed in various domains such as the perceptual speech hashing algorithm [3]. For integrated multimedia services, multimedia compression standards can reduce the data rate dramatically. Various speech compression standards have been recommended by the International Telecommunication Union (ITU) for different applications. Currently, the ITU-T G.723.1 [4-8] and G.729 [9-11] speech codecs are considered the best standards for very low bit rate telephony services. The G.723.1 codec is recommended for H.323 Internet phone systems and the H.324 digital videophone service in public switching telephone network (PSTN) systems [12]. The G.723.1 speech coder has been used on the Internet extensively, for example, in the built-in software NetMeeting in Microsoft windows and other VoIP communication systems. The G.729 codec with lower coding delay and available selection of multiple data rates is the most popular speech codec used in H.323 systems, providing multimedia communication over Internet protocol to achieve guaranteed quality of service for real-time voice, data and video, or any combination of the three, including video telephony [13]. The gateway server processes packet data

transformation between the different speech codecs. Hence, it is obvious that transcoding between G.723.1 and G.729 speech coding standards becomes an important issue for integrating different protocols in the Internet telephony services.

The tandem type approach, namely the decode-then-encode approach, can be considered to achieve communication between the G.723.1 and G.729 speech codecs. We can start with the G.723.1 (G.729) decoding process to reconstruct the compressed speech and then perform G.729 (G.723.1) encoding to complete the transcoding. However, this tandem approach has several problems: firstly, the synthesized speech quality is degraded because the speech signal is encoded and decoded twice using two different speech coders. Quantization errors due to each encoding process accumulate, resulting in degradation of speech quality. Secondly, there will be a longer coding delay and higher computational complexity in speech communications when using two speech coders concurrently. In this paper, we propose a speech transcoding algorithm that directly translates the LSP and open-loop pitch parameters to our target parameters without executing the entire encoding processes. This can significantly reduce the computational load while maintaining good speech quality with no additional look-ahead delay. The application of this transcoding technique can reduce the cost of interoperability over the Internet.

In the existing literature, Neto et al. [14] proposed that pitch delay and fixed-codebook (FCB) index were not changed and were directly mapped from one codec to the other during transcoding between G.729 and IS-641. This is possible since there are many similarities in the structure of the excitation signal. However, if direct mapping is not possible due to dissimilarities in quantizing the excitation signal, the speech quality of the direct mapping approach will seriously degrade. Kang et al. [15] mainly consider transcoding of LSP and gain, and the adaptive codebook (ACB) and the FCB indices are transmitted without modification. Tsai [16] and Ruslan et al. [17] propose transcoding between GSM to G.729, where codecs have the same subframe length. If the subframe lengths of the two codecs are different, the transcoding scheme will not yield better speech quality. Ruslan and Yoon et al. [18-21] proposed a transcoding scheme where the conversion of LSP is performed through the linear interpolation approach and the open-loop pitch parameters are estimated by a pitch smoothing method that executes the cross-correlation criterion. To reduce the G.723.1 ACB search computations, Yoon et al. [20,21] proposed a fast ACB search approach. In addition, to reduce G.723.1 FCB search computations, the approach proposed by Yoon et al. uses a depth-first search algorithm instead of the focused search method originally used in the G.723.1 (5.3 kbit/s) coder. In addition, to reduce search computation, Yoon et al. [20] used the ACELP structure instead of the multi-pulse maximum likelihood quantization (MP-MLQ) structure used in the G.723.1 (6.3 kbit/s) coder, and Yoon's ACELP structure arrangement may ignore a significant excitation pulse at the last track.

In this paper, we propose an efficient transcoding algorithm for G.723.1 and G.729 speech coders. The proposed transcoding algorithm is comprised of four processes: LSP conversion, open-loop pitch conversion, fast ACB search, and fast FCB search, the first two of which use linear interpolation. In the G.723.1 ACB search, we proposed the fast search algorithm based on third-order open-loop pitch gains to predict the candidate gain-vectors of ACB. To reduce ACELP search computational complexity, we propose a fast search algorithm that uses the signal vector $b[n]$ [26] to preselect candidate pulse positions for G.723.1 and G.729 codecs. To reduce MP-MLQ search computational complexity, Lin et al. [22] also provided a candidate pulses approach. In this paper, we modify our previous algorithm to improve the speech quality. The four processes that we have proposed above use techniques that are fully compatible with the ITU-T G.723.1 and G.729 standard speech codecs; these processes do not modify the original protocol of the two standard speech codecs. In other words, the proposed fast search algorithms merely predict candidate pulse positions or gain-vectors.

This paper is organized as follows. In Section 2, the G.723.1 and G.729 speech coding algorithms are briefly reviewed. In Section 3, the transcoding algorithm is proposed to reduce the computational complexity of communications between two speech codecs. The proposed method includes the two fast codebook search algorithms. To verify the efficiency of the proposed techniques, experimental results are presented in Section 4. Finally, we give a conclusion in Section 5.

2. **ITU-T G.723.1 and G.729 Codecs.**

The transcoding algorithm in this paper is associated with ITU-T G.723.1 [4] and G.729 [10]. Conceptually, the G.723.1 and G.729 speech coders adopt the linear predictive coding (LPC) filter to characterize the vocal tract model and use the analogous long-term predictor to depict periodicity of the speech. However, regarding the coded pitch excitation parameter, G.723.1 is based on a fifth-order predictor that is used to predict the quasi-periodic signal of speech, whereas G.729 is designed based on a fractional pitch predictor with 1/3 resolution. The G.729 and G723.1 (5.3 kbit/s) coded stochastic excitations were designed using the focused search and ACELP excitation codebook. G723.1 at 6.3 kbit/s adopts MP-MLQ excitation.

2.1. **ITU-T G.723.1 speech coder.**

ITU-T G.723.1, the standard for multimedia communication speech coders, has two modes with bit rates of 5.3 and 6.3 kbit/s. The coder is based on the principles of linear prediction analysis-by-synthesis coding and attempts to minimize a perceptually weighted error signal. The encoder operates on blocks (30 ms frame) of 240 samples each. Each frame is first divided into four subframes of 60 samples each. In addition, there is a look-ahead of 7.5 ms, so the coder has a 37.5 ms total algorithmic delay. For every 60-sample subframe, a set of tenth order LPC coefficients is computed. The LPC set of the last subframe is converted to LSP parameters, and the LSP set is divided into 3 sub-vectors with dimensions of 3, 3 and 4. The quantization is performed using a predictive split vector quantizer (PSVQ).

The unquantized LPC coefficients are used to construct the short-term perceptual weighting filter, which is used to filter the entire frame speech and to obtain the perceptually weighted speech signal. For every two subframes, the open-loop pitch lag is computed using the weighted speech signal. Every subframe speech signal is then encoded by the ACB and FCB search procedures. The ACB search is performed using a fifth-order pitch predictor to obtain the closed-loop pitch and gains. Finally, the stochastic excitation pulses are approximated by MP-MLQ excitation for high bit rate (6.3 kbit/s), and ACELP for low bit rate (5.3 kbit/s).

Since the speech coder G.723.1 is based on analysis-by-synthesis technology, such a codec structure can achieve high voice quality and low bit rate. However, the shortcoming of this technology is that the encoder requires high computational complexity to search the stochastic codebook. Lee et al. [23] analyze the distribution of computational load for the encoding process of G.723.1 over Samsung's DSP chip in a cost-effective implementation. As Table 1 shows, the MP-MLQ and ACELP codebook search procedures constitute over 55% and 47% respectively of the computations required in the G.723.1 encoding process.

2.2. **ITU-T G.729 speech coder.**

The G.729 codec is based on Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP). The coder operates on a speech frame (block) of 10 ms, which is equivalent to 80 samples at the sampling rate of 8000 Hz. Each block of 10 ms is first divided into two subframes of 40 samples each. There is a 5 ms look-ahead for linear prediction (LP) analysis, resulting in a total 15 ms algorithmic delay.

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

Function description	G.723.1, 6.3 kbit/s	G.723.1, 5.3 kbit/s	G.729, 8kbit/s
LPC	1%	2%	8%
Open-loop pitch	5%	6%	10%
LSP	7%	8%	12%
Filtering	9%	10%	35%
ACB	23%	27%	14%
FCB	55%	47%	21%

For every 10 ms frame, the speech signal is analyzed to extract the parameters of the Code-Excited Linear-Prediction (CELP) coding model. A set of tenth order LPC coefficients are computed using the Levinson-Durbin algorithm. The LPC coefficients for the second subframe are converted to LSP coefficients and are quantized using a predictive two-stage vector quantizer. The unquantized LPC coefficients are used to construct the short-term perceptual weighting filter. After computing the weighted speech signal, an open-loop pitch lag is estimated once per 10 ms frame based on the perceptually weighted speech signal. Next, the ACB and FCB are searched to obtain optimum excitation codevectors. ACB search is performed using a first-order pitch predictor, and a fractional pitch lag with one-third the sample resolution. In the FCB search, the stochastic excitation pulses are modeled using algebraic codebooks with four pulses. Finally, we analyze the distribution of computational load in the encoding process for G.729 using the Visual C++ profile as shown in Table 1.

In the preceding sections, the ACB and FCB of the two speech codecs do not have the same architecture, so the ACB and FCB of codec parameters do not directly translate to target parameters.

3. **The Proposed Transcoding Algorithm.**

The architectures of the tandem system and the proposed transcoder are depicted in Figures 1(a) and 1(b), respectively. For the tandem process, the G.723.1 encoded speech is first decoded by the G.723.1 decoder to obtain decoded speech, which is then compressed by the G.729 encoder to obtain G.729 coded speech. Similarly, the G.729 encoded speech is transformed into G.723.1 coded speech. However, the tandem approach wastes many useful coded speech parameters that exist in the compressed speech in the other format. In this paper, we proposed a transcoding method (depicted in Figure 1(b)) to directly and effectively convert the LSP and open-pitch parameters from G.723.1 (G.729) to G.729 (G.723.1) coded speech. With the proposed transcoding method, we can dramatically reduce the computations required for the encoder in retrieving the LPC and the open-loop pitch parameters.

To convert parameters of the two codecs, we need to solve the frame synchronization problem. For the G.729 coder, each subframe contains 40 samples. As for the G723.1, each subframe contains 60 samples. Unfortunately, the two coders are not the same subframe size. To find translation parameters between G.723.1 and G.729 speech coders, the frame size needs to be synchronized. We compute the least common multiple of the frame size of the two coders. Figure 3 shows the relationship between frames and subframes used in the G.729 and G.723.1 coders. Our transcoding system works in the 240 samples main frame. Each main frame of the transcoding system is equivalent to one frame of G.723.1 and three frames of G.729 coder. With the synchronized main frame, we can begin to explore how to effectively translate the coding parameters to or from the other coder parameters with minimum computations.

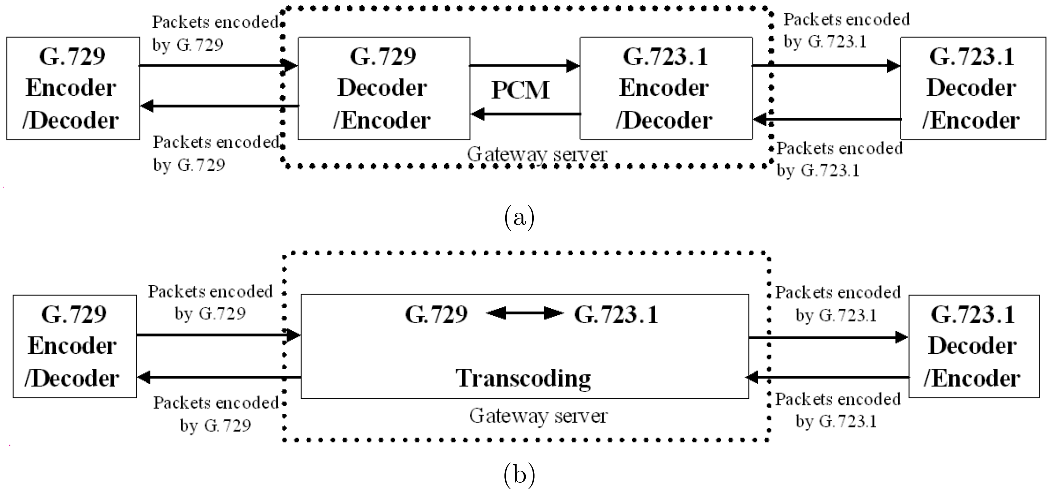

FIGURE 1. (a) System block of the tandem approach, (b) system block of the transcoding method

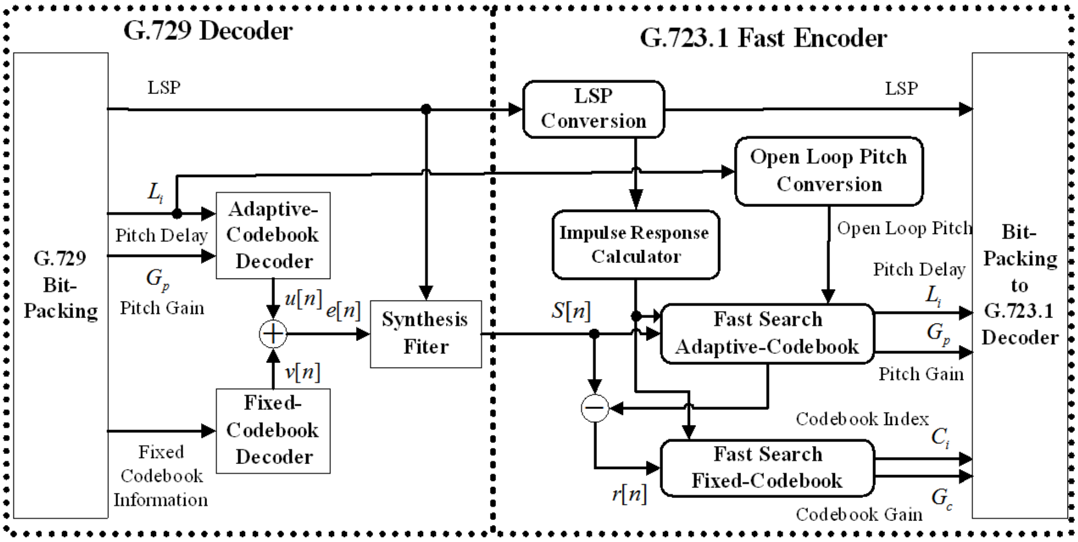

FIGURE 2. Block diagram of the transcoding (from G.729 to G.723.1)

In this paper, we propose a speech transcoding algorithm which directly translates the partial coding parameters to the target parameters without executing the entire encoding processes. Compared with tandem coding, transcoding is more beneficial, as it is a direct translation of one bit stream to the other. Transcoding can prevent degradation of the synthesized speech quality because several source parameters are directly translated in the parametric domain instead of being re-estimated from decoded PCM data. In this respect, the transcoding algorithm is expected to show less distortion than tandem coding. In addition, no extra delay is required, and a reduced computational load is expected.

Finally, to evaluate the performance of the tandem approach and the proposed translation method, we performed objective measurements, which include perceptual evaluation of speech quality (PESQ) (ITU-T Rec.P862) [24], and LPC spectral distortion (*SD*) [25], as well as subjective evaluation using informal mean opinion score (MOS) testing. Twenty speech files were evaluated for speech quality; these files were recorded by 10 males and 10 females in a general environment.

3.1. **Transcoding from G.729 to G.723.1.** Transcoding is performed on the basis that three frames of G.729 are converted to one G.723.1 frame. A block diagram of the proposed transcoding algorithm from G.729 to G.723.1 is shown in Figure 2. The LSP sets of G.729 are converted to those of G.723.1, and are quantized by the G.723.1 encoding algorithm. Quantized LSP sets are converted to LPC coefficients. These coefficients are used to construct the perceptually weighted synthesis filter for each subframe. The open-loop pitch in the G.723.1 encoder is obtained using linearly interpolated G.729 decoded pitch parameters. Afterwards, ACB and FCB parameters are found by filtering the excitation signal through the perceptually weighted synthesis filter. To reduce computational complexity of the ACB and FCB search, we proposed two fast search schemes for each. Finally, the parameters of G.723.1 are encoded to yield the bit stream, which transmitted to the G.723.1 decoder.

3.1.1. *LSP conversion using a linear interpolation.* Figure 3 shows the LSP conversion scheme. One G.723.1 frame size can be divided into three G.729 frames. Thus, three sets of LSP parameters obtained from the G.729 coding parameters should be interpolated to four sets of LSP parameters for the G.723.1 encoder. Let $L_j^A[i]$ be the LSP coefficients of the jth G.729 frame for $j = 1$, 2 and 3, and $L_j^B[i]$ be the LSP coefficients of the jth G.723.1 subframe for $j = 1$, 2, 3 and 4. The relationship of LSP parameters' interpolation is given by

$$\left.\begin{aligned} L_1^B[i] &= L_1^A[i] \\ L_2^B[i] &= \frac{1}{3} \times (L_1^A[i] + 2 \times L_2^A[i]) \\ L_3^B[i] &= \frac{1}{3} \times (2 \times L_2^A[i] + L_3^A[i]) \\ L_4^B[i] &= L_3^A\ [i] \end{aligned}\right. \qquad i = 1, 2, \ldots, 10, \tag{1}$$

where $L_j^A[i]$ and $L_j^B[i]$ are the LSP parameters of G.729 and G.723.1, respectively. The quantized L_4^B set is transmitted to the decoder. Thus, the high computational complexity of calculating the LP coefficients in the G.723.1 encoder can be eliminated.

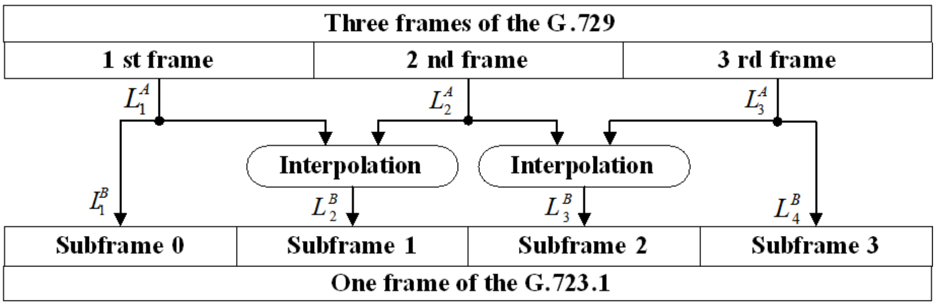

FIGURE 3. LSP conversion using linear interpolation (from G.729 to G.723.1)

TABLE 2. Spectral distortion (from G.729 to G.723.1)

Method	Average *SD*	Distribution		
		< 2 dB	2-4 dB	> 4 dB
Tandem	2.376 dB	78.351%	21.306%	0.344%
Transcoding	1.526 dB	96.907%	3.093%	0%

To evaluate the performance of the tandem approach and the proposed LSP conversion method, we measured the spectral distortion, as shown in Table 2. The decoded LPC coefficients of G.723.1 were used as a reference. The spectral distortion of the proposed transcoding method is much less than that of the tandem approach. Moreover, the spectral distortion of the proposed transcoding method is usually less than 2 dB. In addition, cases of spectral distortion being larger than 4 dB are rare. We also compared the LPC spectra of the tandem approach and the proposed method to that of the original G.723.1 method. Figure 4 shows the LPC spectra in the voice region of the speech signal in which the LPC spectrum of G.723.1 is also shown as a reference. The original speech was first processed by the G.723.1 encoder, and then decoding was performed to obtain the decoded LPC coefficients, which were used as a reference to evaluate *SD* and LPC spectra. It is notable that the LPC spectrum of tandem approach indicates a larger spectral distortion than

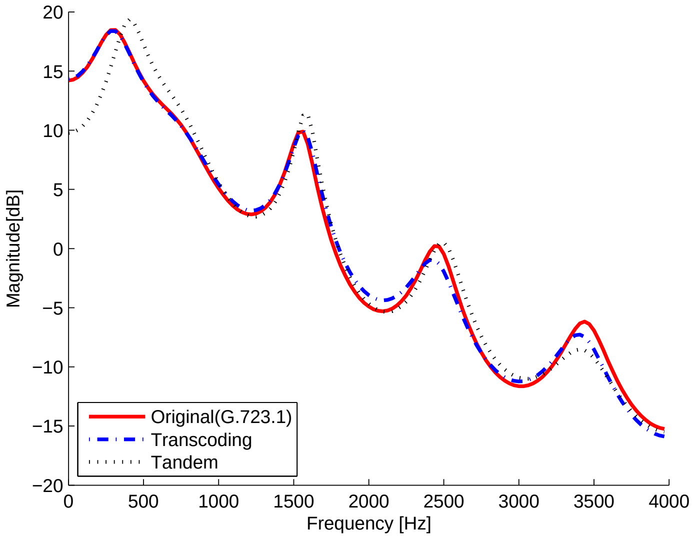

FIGURE 4. The comparison of the LPC spectra (from G.729 to G.723.1)

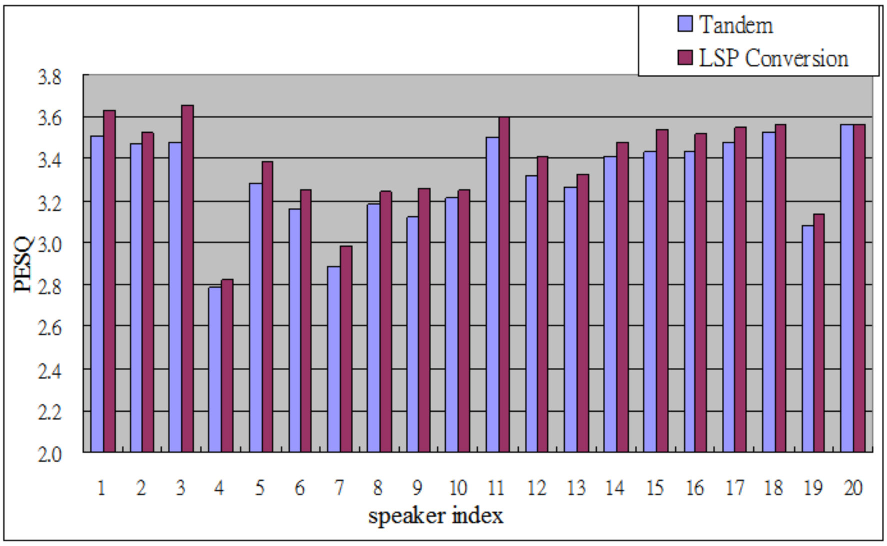

FIGURE 5. PESQ of the tandem approach and the LSP conversion method (from G.729 to G.723.1)

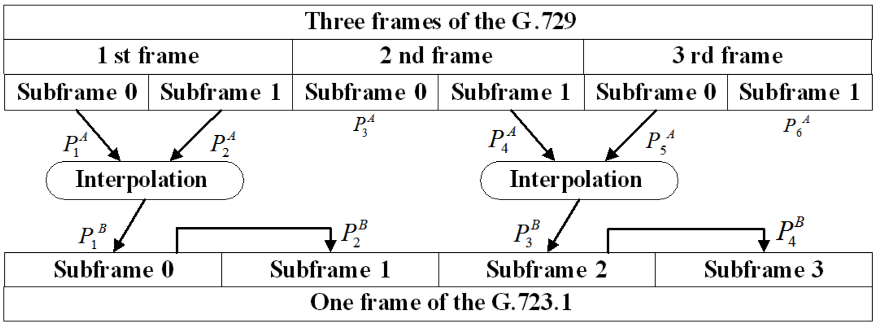

Figure 6. Open-loop pitch conversion using linear interpolation (from G.729 to G.723.1)

that the proposed method, and therefore the proposed LSP conversion technique can provide better speech quality than the tandem approach. In addition, we also compared the PESQ of the tandem approach and the proposed LSP conversion method, as shown in Figure 5. It is observed that the speech quality of the proposed transcoding scheme is better than that of the tandem approach, as the average PESQ score of the former is about 0.0803 better than that of the latter and higher PESQ and lower *SD* imply better speech quality.

3.1.2. *Open-loop pitch conversion using a linear interpolation.* Normally, pitch prediction should start from an open-loop search, which requires computation of the autocorrelation function of perceptually weighted speech signals. From the open-loop pitch solution, we should perform a precise closed loop search, which has high computation requirements. The aim here is to effectively utilize the received pitch information of one coder to estimate a pitch parameter for another coder. In the G.723.1 coder, we need to perform two open-loop pitch estimation processes per frame. Thus, we need to compute an open-loop pitch predictor for every two subframes. Due to the synchronized frames shown in Figure 6, there are four 60-sample subframes in the G.723.1 coder and six 40-sample subframes in the G.729 coder in every 240 samples. To reduce computational complexity, the open-loop pitch in the G.723.1 encoder is directly obtained from the G.729 decoded pitch lag parameters. The transformation can be done using the formula:

$$\begin{aligned} P_1^B &= \frac{1}{3} \times (2 \times P_1^A + P_2^A) \\ P_3^B &= \frac{1}{3} \times (2 \times P_4^A + P_5^A) \end{aligned} \tag{2}$$

where P_1^A, P_2^A and P_4^A, P_5^A are decoded pitch lag relative to subframes of the G.729 respectively. The P_1^B and P_3^B parameters are open-loop pitch lag for subframe 0 and subframe 2 of the G.723.1 encoder, respectively. Therefore, the open-loop search process procedure in the G.723.1 encoder can be omitted. However, Yoon et al. [20] proposed an open-loop pitch smoothing approach to obtain the open-loop pitch parameters of G.723.1 and Yoon's proposed approach requires higher computational complexity than the linear interpolation method.

In order to compare the performance of the tandem approach with that of the proposed pitch conversion method, we measured the PESQ and open-loop pitch contour. As shown in Figure 7, the open-loop pitch contour of the proposed method matches well with that of the original G.723.1. In addition, we implemented the LSP and pitch conversion schemes in the G.723.1 encoder. Figure 8 shows the integral PESQ. It is observed that the average

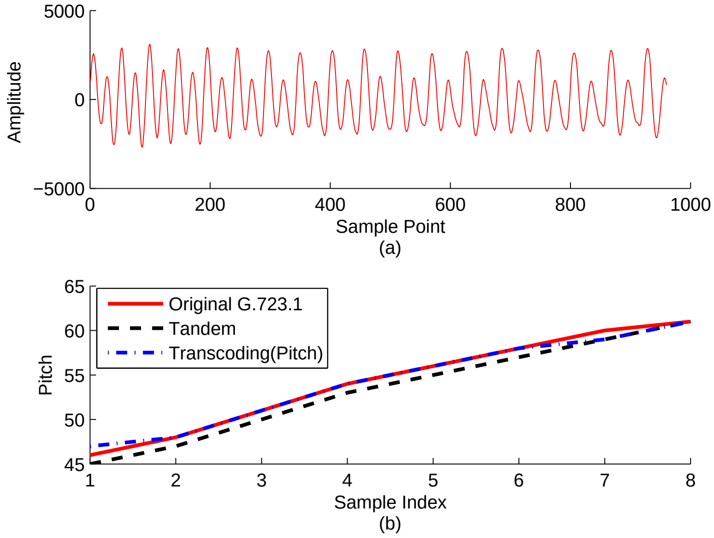

Figure 7. Comparison of open-loop pitch contour: (a) voiced speech segment, (b) the estimated open-loop pitch contour

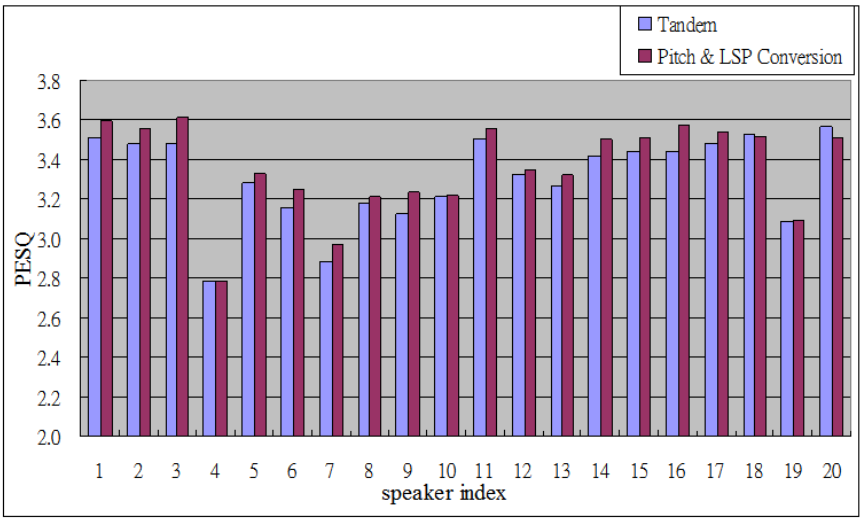

Figure 8. PESQ of the tandem approach and the integrated LSP and Pitch conversion method (from G.729 to G.723.1)

speech quality of the proposed transcoding scheme is about 0.05595 higher than that of the tandem approach.

3.1.3. *Fast adaptive-codebook search.* The adaptive-codebook (ACB) search in G.723.1 uses a fifth-order pitch predictor, and estimates the pitch lag and gains simultaneously. The pitch predictor gains are vector quantized using two gain-codebooks (GB) with 85 or 170 entries for the high bit rate, 170 entries for the low bit rate, and each gain vector has 20 elements. For example, 170 entries GB for the low bit rate, to obtain optimal closed-loop pitch lag and the related gains vector, must search 4×170 gain-vectors of the ACB for subframes 1 and 3, whereas for subframes 0 and 2 also search 3×170 gain-vectors. Thus, the ACB search in G.723.1 requires heavy computation. On average, every subframe requires 3.5 iterations to search the 85 or 170 entries of the gain-codebook. As a result, every subframe on average requires searching $3.5 \times (170 + 85)/2 = 446$ gain-vectors. It is observed that the efficiency of the G.723.1 encoding speech signals is improved by reducing the amount of the searched gain-vectors. We propose a fast ACB search algorithm to reduce the computational complexity. Our scheme utilizes residual signals and third-order open-loop pitch gains to preselect candidate gain-vectors of the ACB, and this estimation was performed before the original ACB search procedure. Finally, the G.723.1 ACB coding process only searches these candidate gain-vectors. The flow chart of the proposed scheme is shown in Figure 9. Firstly, the target signal $S[n]$ was filtered by a $1/H(z)$ filter to generate the excitation signal $E[n]$, and generate the excitation signal $e'_k[n]$ using open-loop pitch lag L and ACB gains. The function is given by

$$E[n] = S[n] - \sum_{i=1}^{10} a[i]S[n-i], \quad 0 \le n \le 59 \tag{3}$$

$$e'_k[n] = \sum_{j=-1}^{j=1} \beta_{kj} e[n-L+j], \quad 0 \le n \le 59 \tag{4}$$

where $a[i]$ are LPC coefficients, $e[n]$ is the previous excitation signal and β_{kj} are ACB gain-vectors. The open-loop third-order pitch gain-vectors is searched by using Equation (5)

$$MSE_k = \sum_{n=0}^{59} (E[n] - e'_k[n])^2, \quad 0 \le n \le 59, \quad 0 \le k \le 169 \text{ or } 84 \tag{5}$$

We adopt minimum squared error (MSE_k) as the criterion to estimate M candidate gain-vectors from ACB 170 or 85 gain-vectors, where k is the index of ACB. The above

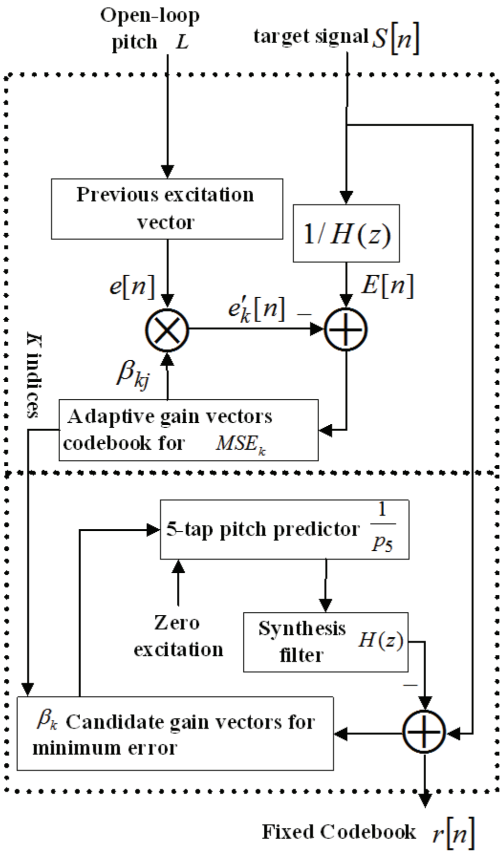
Open-loop pitch L
target signal $S[n]$
Previous excitation vector
$1/H(z)$
$e[n]$
$E[n]$
$e'_k[n]$
–
β_{kj}
K indices
Adaptive gain vectors codebook for MSE_k
5-tap pitch predictor $\frac{1}{P_5}$
Zero excitation
Synthesis filter $H(z)$
–
β_k Candidate gain vectors for minimum error
Fixed Codebook $r[n]$

FIGURE 9. Flow chart of the proposed fast search algorithm

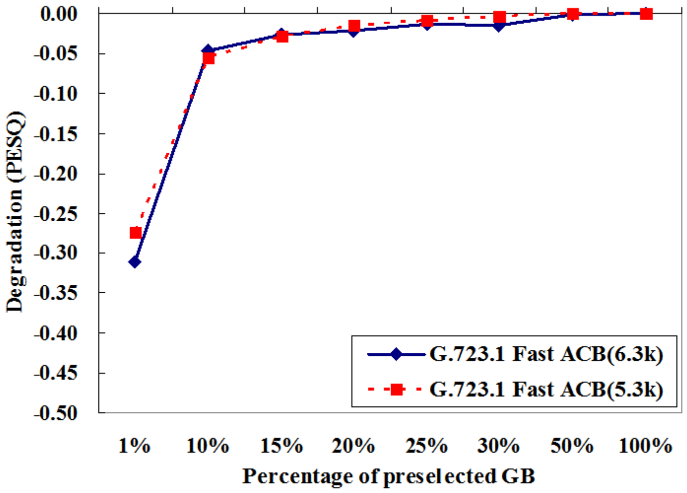

Figure 10. Average degradation PESQ with regard to different percentages of preselected GB

process is an open-loop search, which requires less computation than close-loop search. In other words, close-loop pitch gains coding only searches these M candidate gain-vectors. We estimate speech quality relative to preselected candidate gain-vectors in the experiment, with results shown in Figure 10. We observe that the number of candidate gain-vectors from experimentation produces $GB \times 15\% = M$ gain-vectors and can achieve optimality in terms of speech quality and computational complexity.

For example, 15% of GB with smaller value MSE_K using Equation (5) were preselected as the candidate gain-vectors. It should be noted that the G.723.1 ACB coding process only searches these candidate gain-vectors. Consequently, the proposed algorithm only tests $3.5 \times 0.15 \times (170 + 85) \div 2 = 67$ candidate gain-vectors for every subframe. Therefore, the proposed fast search algorithm can reduce the computational complexity by about 85% in the ACB search. It should be noted that under the fast ACB search approach proposed by Yoon and Jung et al. [20,21] 85 gain-vectors need to be tested for every subframe. However, the preprocessing for deciding the candidate gain-vectors requires an extra computational load of about 3.5% [22]. The preselected 15% gain-vectors from GB are used in the experiment, and the results show that the average of the PESQ score is degraded slightly, by only 0.027, relative to the original G.723.1 full search procedure. However, the proposed method can dramatically reduce the computational complexity by about 81.5% with perceptually negligible degradation. Experiment results are as shown in Figure 11.

3.1.4. *Fast fixed-codebook search.* (for 6.3 kbit/s MP-MLQ). After the short-term analysis and long-term prediction, the weighted residual signal, $r[n]$, is obtained as a new target signal for stochastic excitation processing. The stochastic excitation search procedure, which performs estimation and quantization for the target vector, involves the determination of pulse position and amplitude. For the 6.3kbit/s bit rate, MP-MLQ excitation signal is used, and the coder is based on the analysis-by-synthesis technology. Such a codec structure can achieve high voice quality and low-bit rate; the shortcoming of this technology is that the encoder requires much computational complexity.

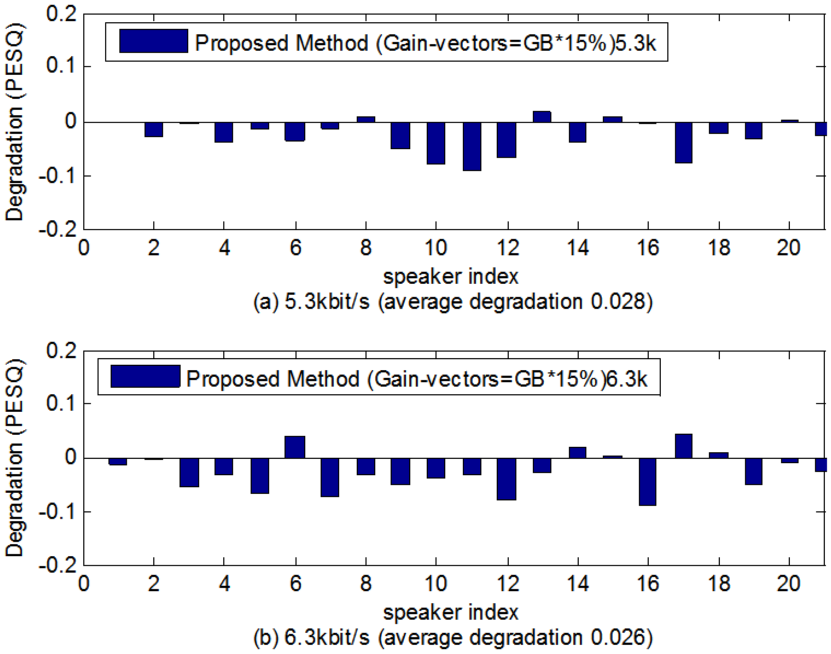

Figure 11. The degradation of PESQ of the proposed method compared with the original ACB full search

To achieve a good approximation of the target signal, $r[n]$, the encoding process by $r'[n]$ is given as

$$r'[n] = \sum_{j=0}^{n} h[j] \cdot v[n-j], \quad 0 \le n \le N-1 \tag{6}$$

where N is the subframe length and $v[n]$ denotes the excitation to the synthesis filter $h[n]$. $v[n]$ can be expressed as [4]

$$v[n] = G \sum_{k=0}^{M-1} \alpha_k d[n-m_k], \quad 0 \le n \le N-1 \tag{7}$$

where G is the gain factor, m_k denotes the excitation pulse position with $\alpha_k = \pm 1$ and M is the number of pulses, which is 6 for even

subframes and is 5 for odd subframes. There is a restriction on pulse positions in the G.723.1 coder. The positions can either be all odd or all even. Consequently, the optimization estimates the unknown parameters G, $\{\alpha_k\}$ and $\{m_k\}$ for $k = 0, 1, \ldots, M-1$, such that they minimize the mean square of the error signal, $err[n]$:

$$err[n] = r[n] - r'[n] = r[n] - G \sum_{k=0}^{M-1} \alpha_k h[n-m_k] \tag{8}$$

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

$$d[j] = \sum_{n=j}^{N-1} r[n] \cdot h[n-j], \quad 0 \le j \le N-1 \tag{9}$$

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

$$G_{\max} = \frac{\max\{|d[j]|\}_{j=0\dots N-1}}{\sum\limits_{n=0}^{N-1} h[n] \cdot h[n]} \tag{10}$$

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

The optimal combination of pulse positions and gain is encoded. $2 \times C_M^{30}$, $M = 6, 5$ combinatorial coding is used for the pulse positions. 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 will help to improve encoder efficiency.

The signal vector $b[n]$ is used in AMR to search the algebraic codebook. Previously, we proposed a fast search algorithm to reduce computational complexity of the MP-MLQ search algorithm [22]. In this algorithm, we used the signal vector $b[n]$ to preselect candidate pulse positions. To further improve the speech quality, we modified our previous algorithm in this paper. The flow chart of the proposed scheme is shown in Figure 12. Firstly, the target signal, $r[n]$, was filtered by the $A(z)$ filter to generate the excitation signal $res_{LTP}[n]$ for each subframe, where $A(z)$ filter is defined as

$$A(z) = 1 - \sum_{i=1}^{10} a[i] z^{-i} \tag{11}$$

where $a[i]$ are LPC coefficients. The pulse-position likelihood-estimate vector, $b[n]$ [26] is defined as

$$b[n] = \frac{|res_{LTP}[n]|}{\sqrt{\sum\limits_{i=0}^{N-1} res_{LTP}[i] \cdot res_{LTP}[i]}} + \frac{|d[n]|}{\sqrt{\sum\limits_{i=0}^{N-1} d[i] \cdot d[i]}}, \quad 0 \le n \le N-1 \tag{12}$$

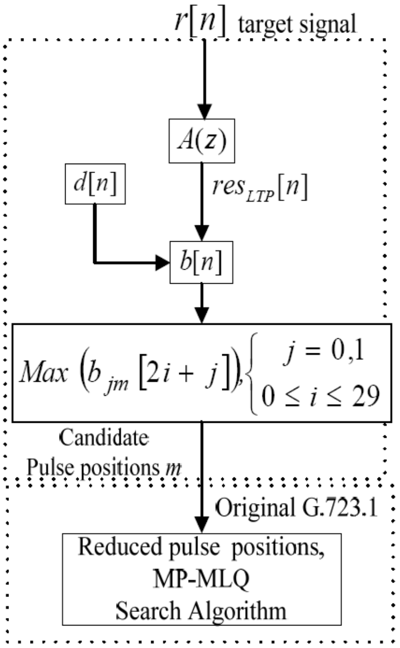
$r[n]$ target signal
$A(z)$
$d[n]$
$res_{LTP}[n]$
$b[n]$
$Max\left(b_{jm}[2i+j]\right), \begin{cases} j = 0,1 \\ 0 \le i \le 29 \end{cases}$
Candidate Pulse positions m
Original G.723.1
Reduced pulse positions, MP-MLQ Search Algorithm

Figure 12. Flow chart of the proposed fast search algorithm for MP-MLQ

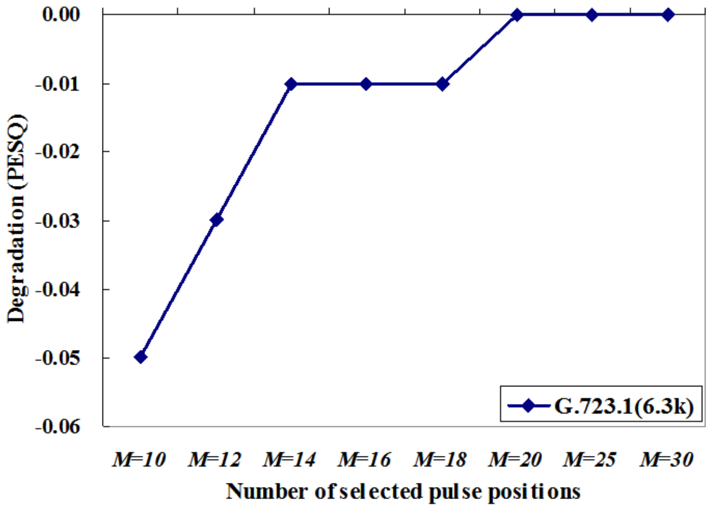

Figure 13. PESQ estimation for different pulse number M (original MP-MLQ $M = 30$)

where N is the subframe length, to preselect M candidate pulse positions with larger value $b_{jm}[n]$ of odd positions ($j = 1$) and even positions ($j = 0$) for every subframe, respectively, where subscript m denotes the selected excitation pulse positions. In other words, the signal vector $b[n]$ is used only to preselect candidate pulse positions. The estimate candidate pulse position process was performed before the original standard MP-MLQ search procedure. It is noted that the MP-MLQ coding process only searches these $2M$ ($2M < N$) candidate pulse positions for every subframe. We analyzed the average degradation of PESQ relative to the number of candidate pulse positions in the experiment, the results of which are shown in Figure 13. It is observed that preselecting $M = 14$ positions can achieve the optimality in terms of speech quality and computational complexity. For example, preselecting candidate pulse positions $M = 14$, the number of combinations of all possible pulse positions will be reduced from $1187550(2 \times C_6^{30})$ to $6006(2 \times C_6^{M})$ for even subframes. In addition, for odd subframes the number of possible combinations of the positions will be reduced from 285012 to 4004.

The computational complexity of MP-MLQ coding is therefore significantly reduced by using our proposed method. However, the preprocessing for deciding the candidate pulse positions require extra computational load of about 3.5% [22]. We preselected $M = 14$ pulse positions in the experiment, and the results show that the average degradation of the PESQ score is about 0.009 relative to the original search procedure. However, the proposed method reduces the computational complexity by about 95.8%, with perceptually negligible degradation. In order to reduce the search computation requirements for this stage, Yoon et al. [20] proposed an approach that applies ACELP search instead of the MP-MLQ search originally implemented in G.723.1, which reduced computation requirements by about 80%. Experimental results are shown in Figure 14, comparing the true PESQ values of the proposed method with original MP-MLQ search.

3.1.5. *Fast fixed-codebook search (for G.723.1, 5.3 kbit/s ACELP).* In the ACELP coders, we need to find the four pulses, which include their positions and amplitudes, to synthesize the best matched of the target signal. The candidate pulse positions are partitioned into 4 tracks, t_0, t_1, t_2, t_3 as Table 3 shown the positions for the G.723.1 ACELP codebook.

The nest-loop search scheme is the optimal method to discover the solution of pulse positions, and this method must search the total possible pulse position combinations,

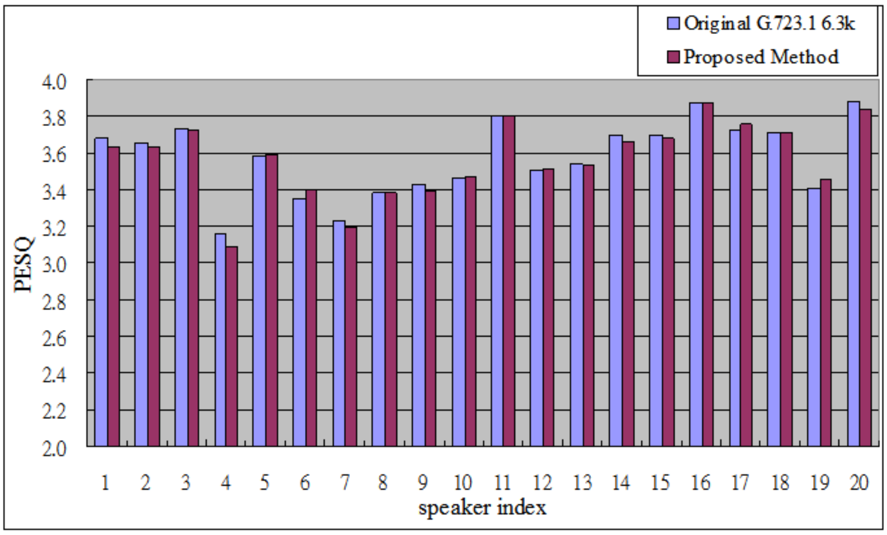

Figure 14. PESQ of the original MP-MLQ search and the proposed method ($M = 14$)

Table 3. The G.723.1 ACELP excitation codebook

	Sign	Positions
t_0	±1	0(1), 8(9), 16(17), 24(25), 32(33), 40(41), 48(49), 56(57)
t_1	±1	2(3), 10(11), 18(19), 26(27), 34(35), 42(43), 50(51), 58(59)
t_2	±1	4(5), 12(13), 20(21), 28(29), 36(37), 44(45), 52(53)
t_3	±1	6(7), 14(15), 22(23), 30(31), 38(39), 46(47), 54(55)

6272 loops ($8 \times 8 \times 7 \times 7 \times 2$) in the G.723.1, so it can achieve the best speech quality. However, the shortcoming of this technology is that it requires much computational complexity to synthesize the best match of the target signal. To reduce the nest-loop search computational complexity, there will be simplified methods that set some additional restrictions to decrease the possible combinations of pulses. The ITU-T and the ETSI committees suggest the focused search method for the G.723.1. To further control the search, the number of times the last loop is entered (for the 4 subframes) is not allowed to exceed 600. Hence, the maximum number of possible position combinations becomes 1498 ($8 \times 8 \times 7 + 150 \times 7$), and if the depth-first search process is used, 300 combinations will also be required for the G.723.1 encode every subframe. In order to reduce computational complexity of the original G.723.1 focused search, we propose a fast search method using the signal vector, $b[n]$ to preselect M candidate pulse positions with larger value, $b_t^j[n]$ for every track of odd positions ($j = 1$) and likewise for even positions ($j = 0$). Next, a depth-first search is implemented in the proposed method, which is used in the G.729A FCB search. In other words, the depth-first search algorithm combining signal vector $b[n]$ is implemented to further reduce computational complexity. The $b[n]$ signal vector is defined as shown in Equation (12). Similarly, the preprocessing for deciding the candidate pulse positions require extra computational load of about 3.5% [22]. The flow chart of the proposed scheme is shown in Figure 15. It should be noted that the depth-first search process only searches M candidate pulse positions for every track. We estimate speech quality relative to preselected M candidate pulse positions in our experiment. Figure 16 shows the average degradation PESQ versus the number of candidate pulse positions relative to computational complexity for the proposed method

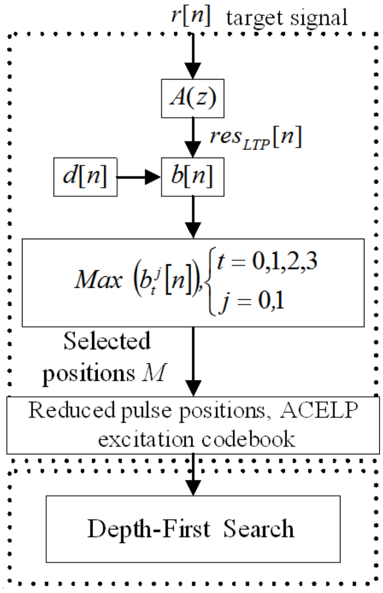
$r[n]$ target signal
$A(z)$
$res_{LTP}[n]$
$d[n]$
$b[n]$
$Max\ (b_t^j[n]),\begin{cases} t=0,1,2,3 \\ j=0,1 \end{cases}$
Selected positions M
Reduced pulse positions, ACELP excitation codebook
Depth-First Search

FIGURE 15. Flow chart of the proposed fast search algorithm for G.723.1ACELP

compared with the original ACELP focused search. We observe that the number of candidate pulse positions for every track from experimentation produces $M = 4$ positions and can achieve an optimum in terms of speech quality and computational complexity. Finally, the number combinations of search pulse positions will be reduced from 1498 to 96 $(4 \times (2 \times 4 + 4 \times 4))$ for every subframe. We preselected pulse positions $M = 4$ for every track in the experiment, with the results showing that the average of PESQ score is degraded slightly, by only 0.01, relative to the original focused search. However, the proposed method reduced the computational complexity about 90.09% with perceptually negligible degradation. Figure 17 shows the true PESQ values in this experiment. For this stage, Yoon et al. [20] only employed a depth-first search approach, and their approach required testing 256 position combinations for every subframe.

3.2. **Transcoding from G.723.1 to G.729.** For the case of speech communication from G.723.1 to G.729, the proposed method consists of LSP conversion, open-loop pitch conversion, and fast FCB search. Transcoding is executed on the basis of G.723.1 frame length, and therefore one G.723.1 frame is converted to three G.729 frames. The LSP conversion and open-loop pitch conversion for these schemes and structures are similar to that for G.729 to G.723.1. A block diagram of the proposed transcoding algorithm from G.723.1 to G.729 is shown in Figure 18. The FCB of G.729 is based on ACELP structure using the focused search method, but the ACELP structure is not similar to that of the G.723.1. However, to achieve optimality in terms of speech quality and computational complexity, we will propose a fast search scheme, which is similar to that in Section 3.1.5 earlier.

3.2.1. *LSP conversion using a linear interpolation.* A linear interpolation technique is used to translate the LSP parameters. The frame length of G.723.1 is three times than that of the G.729. The subframes of the G.723.1 are named L_1^B, L_2^B, L_3^B and L_4^B and the frames of the G.729 are named L_1^A, L_2^A and L_3^A. Four sets of LSP parameters obtained from the G.723.1 coding parameter should be interpolated to three sets of LSP parameter for the G.729 encoder. Figure 19 shows the LSP conversion scheme, which can be expressed

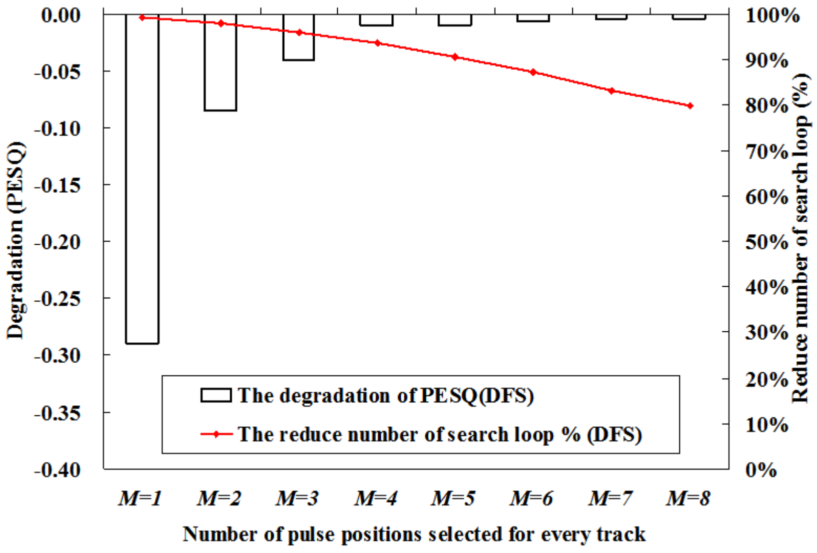

FIGURE 16. Average degradation PESQ versus computation with regard to different number M of pulse positions (original ACELP every track $M = 8$)

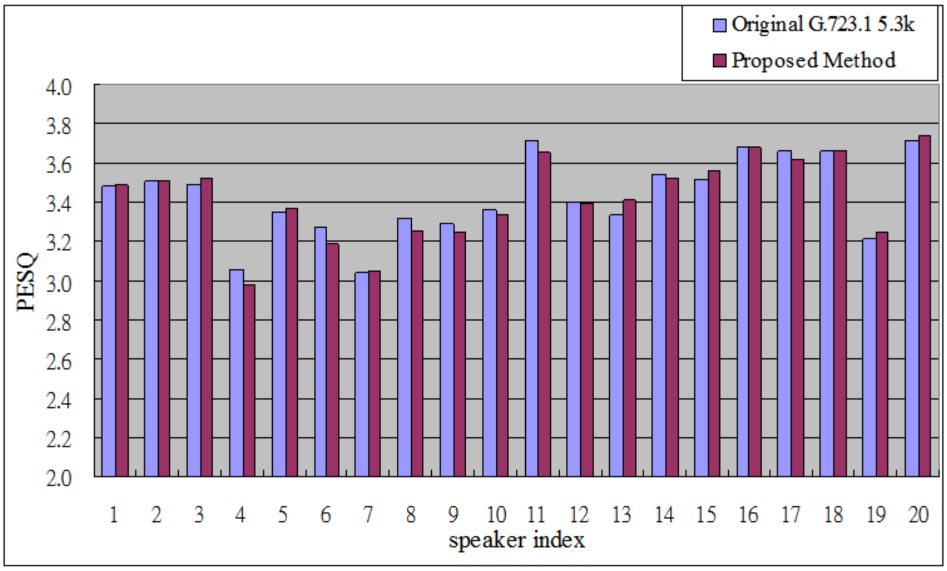

FIGURE 17. PESQ of the original ACELP search and the proposed fast ACELP search ($M = 4$)

by

$$\begin{aligned} L_1^A[i] &= \frac{1}{4} \times (3 \times L_1^B[i] + L_2^B[i]) \\ L_2^A[i] &= \frac{1}{2} \times (L_2^B[i] + L_3^B[i]) \qquad\qquad i = 1, 2, \ldots, 10. \\ L_3^A[i] &= \frac{1}{4} \times (L_3^B[i] + 3 \times L_4^B[i]) \end{aligned} \tag{13}$$

Thus, the high computational complexity of the LP coefficients calculation in the G.729 encoder can be eliminated. As for spectral distortion estimation, the results of which are shown in Table 4, the spectral distortion of the proposed LSP conversion is much less

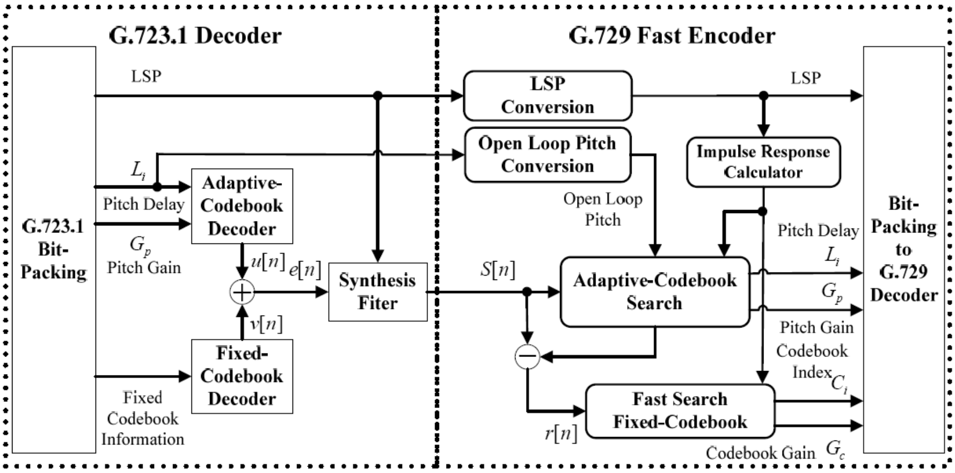

FIGURE 18. Block diagram of the transcoding (from G.723.1 to G.729)

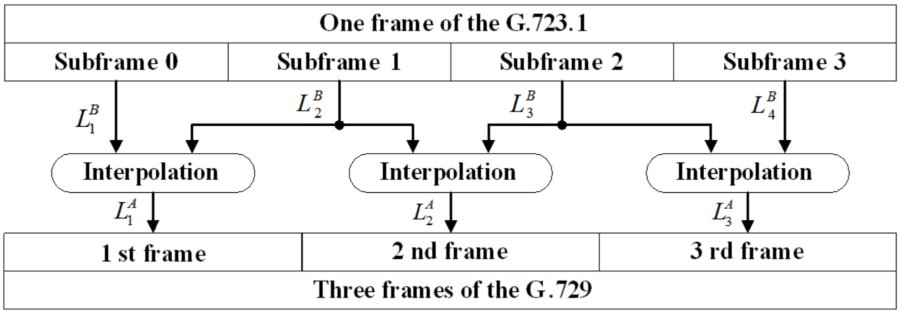

FIGURE 19. LSP conversion using linear interpolation (from G.723.1 to G.729)

TABLE 4. Spectral distortion (from G.723.1 to G.729)

Method	Average *SD*	Distribution		
		< 2 dB	2-4 dB	> 4 dB
Tandem	2.912 dB	60.680%	31.298%	8.015%
Transcoding	1.571 dB	94.275%	5.725%	0%

than that of the tandem approach. Moreover, the spectral distortion of the proposed transcoding method is usually less than 2 dB. In addition, cases of spectral distortion being larger than 4 dB are rare. As shown in Figure 20, the LPC spectrum of the proposed method matches well with the reference spectrum obtained from the original G.729 encoder compared to that of the tandem approach. The original speech was first processed by the G.729 encoder, and then decoding was performed to obtain the decoded LPC coefficients, which were used as a reference to evaluate *SD* and LPC spectra. We also compared the PESQ of the tandem approach and the proposed LSP conversion scheme, as shown in Figure 21.

3.2.2. *Open-loop pitch conversion using a linear interpolation.* After the LSP conversion, the open-loop pitch for each frame of G.729 is estimated. Due to frames synchronization, there are four G.723.1 subframes, whose pitches are denoted as P_1^B, P_2^B, P_3^B and P_4^B. The three open-loop pitch parameters of the G.729 frame are obtained by using linearly interpolating G.723.1 decoded pitch lag parameters. Figure 22 shows the open-loop pitch conversion procedure, which can be expressed by

$$\begin{aligned} P_1^A &= \frac{1}{4} \times (3 \times P_1^B + P_2^B) \\ P_3^A &= \frac{1}{2} \times (P_2^B + P_3^B) \\ P_5^A &= \frac{1}{4} \times (P_3^B + 3 \times P_4^B) \end{aligned} \tag{14}$$

where P_j^A denotes the open-loop pitch lag of the jth G.729 subframe.

The P_1^A, P_3^A and P_5^A parameters are open-loop pitch lags for the first, second and third frame of the G.729 encoder, respectively. Therefore, G.729 encoder can avoid the open-loop search procedure. For this stage, Yoon et al. [20] proposed an open-loop pitch smoothing technique, which requires much higher computational complexity than the linear interpolation method.

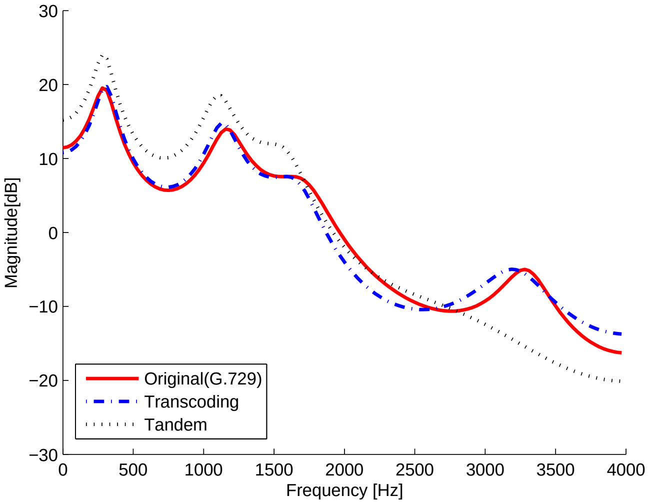

Figure 20. The comparison of the LPC spectra (from G.723.1 to G.729)

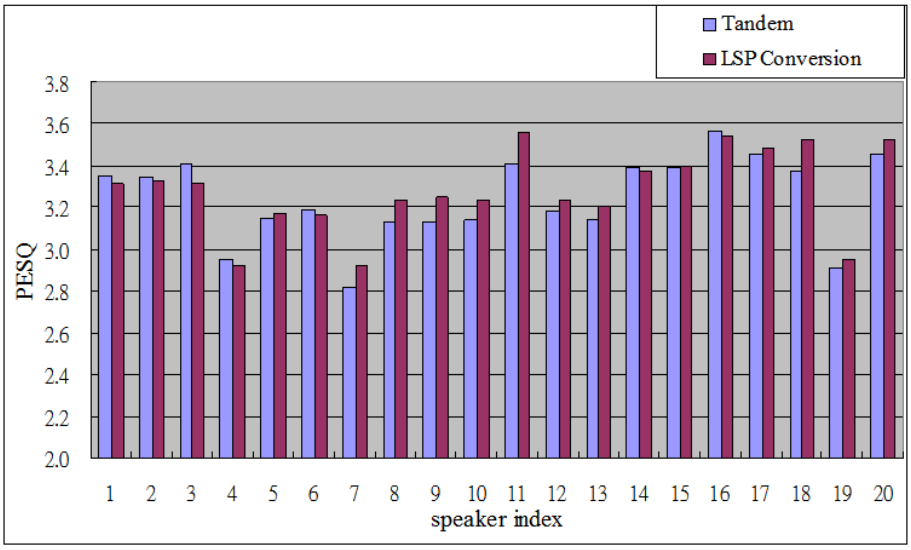

Figure 21. PESQ of the tandem approach and the LSP conversion method (from G.723.1 to G.729)

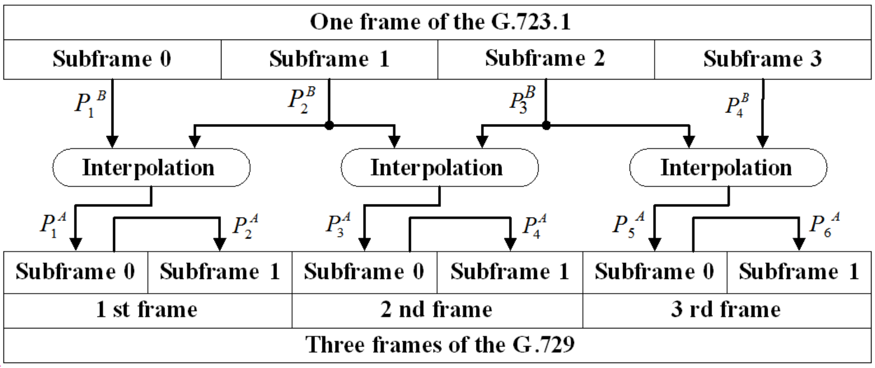

FIGURE 22. Open-loop pitch conversion using linear interpolation (from G.723.1 to G.729)

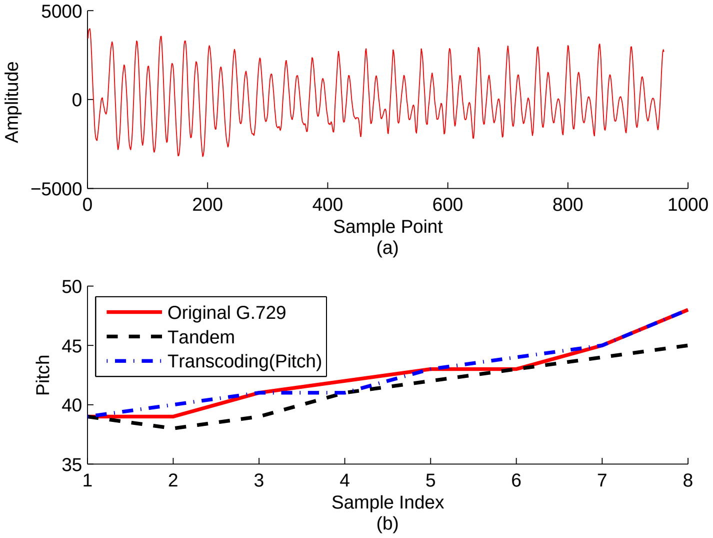

FIGURE 23. Comparison of open-loop pitch contour: (a) voiced speech segment, (b) the estimated open-loop pitch contour

To evaluate the performance of the tandem approach and the proposed pitch conversion method, we estimate open-loop pitch contour and PESQ, respectively. As shown in Figure 23, the open-loop pitch contour of the linear interpolation method matches well with the pitch of the original G.729. In addition, we simultaneously implement LSP and pitch conversion in the G.729 encoder, and Figure 24 shows the integral PESQ. Observations show that the average speech quality of the proposed transcoding scheme is better than that of the tandem approach. Moreover, the proposed method improves the average PESQ about 0.0241 more than that of the latter.

3.2.3. *Fast fixed-codebook search.* The focused search scheme was also used in the G.729 ACELP codebook coding, which requires 1952 ($8 \times 8 \times 8 + 90 \times 16$) loops to search four pulses for every subframe. To further reduce computation of focused search, we propose a fast search scheme, which is similar to the one described in Section 3.1.5. The proposed method uses the signal vector $b[n]$ to preselect M candidate pulse positions; these M candidate pulse positions with larger values of $b[n]$ are preselected for every track, and these M candidate pulse positions are then coded by depth-first search. However, previous research [20] does not describe the fast FCB search approach of G.729A. Figure 25 shows

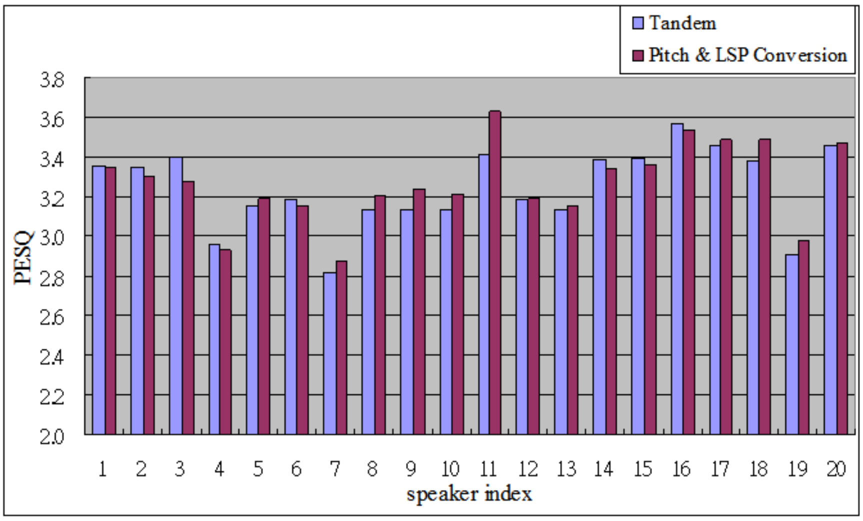

FIGURE 24. PESQ of the tandem approach and the integrated LSP and Pitch conversion method (from G.723.1 to G.729)

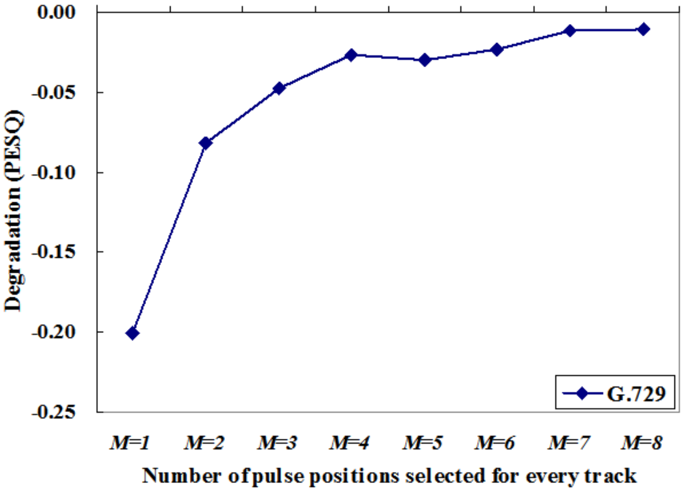

FIGURE 25. The average degradation PESQ with regard to different number M of pulse positions (using depth-first search .original ACELP every track $M = 8$)

the average degradation PESQ relative to number of candidate pulses for the proposed method compared with the original G.729 ACELP focused search. We observe that the number of candidate pulse positions for every track from experimentation produces $M = 4$ positions and can achieve an optimum in terms of speech quality and computational complexity. Finally, the number of combinations of the search pulse positions will be reduced from 1952 to 96 for every subframe. Experimental results show that the average degradation of PESQ score is about 0.02655 relative to the original G.729 focused search. However, the proposed method reduces the computational complexity by about $((1952 - 96)/1952) - 3.5\% = 91.58\%$ with perceptually negligible degradation.

4. **Overall Performance Analysis.** In this paper, we proposed an efficient method of converting speech codec formats between the G.729 and G.723.1 speech codecs. To evaluate the overall performance of the proposed transcoding algorithm and the tandem

approach, subjective preference tests are performed together with objective speech quality evaluation and computational complexity analysis. Subjective speech quality is evaluated via an A–B preference test, and an objective speech quality measure PESQ is used.

Twenty speech files are tested for speech quality evaluation. In our experiments, the LSP, open-pitch conversion, and fast FCB search algorithms were simultaneously implemented in both the G.723.1 and G.729 encoders, and the fast ACB search algorithm was also simultaneously implemented in the G.723.1 codec. It must be noted that the average PESQ score can be considered as an evaluation of the overall performance.

4.1. **Objective speech quality evaluation.** For objective evaluations PESQ, the measurement results are summarized in Tables 5 and 6, and testing results show that the proposed transcoding scheme produces speech quality equivalent to that of tandem approach.

4.2. **Subjective speech quality evaluation.** To verify objective results of the PESQ measurements, a simple and informal MOS assessment is also offered in this paper. We implement a subjective quality measurement called the A–B test. 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 Tables 7 and 8, and test results show that the average speech quality of the proposed transcoding scheme is better than that of the tandem approach, slightly. Accompanying the subjective tests above, we also provide readers the decoded sound files on the website at http://faculty.stut.edu.tw/∼rslin/listening.htm for subjective listening.

4.3. **Computational complexity analysis.** For the tandem approach, each speech frame was encoded and decoded twice by the two different speech coders. Therefore, the tandem approach required very high computational complexity with an additional look-ahead coding delay. However, we propose a speech transcoding algorithm that directly translates coding parameters to target parameters without executing the encoding process. In other words, the calculations of LPC, LSP and open-loop pitch conversion in

Table 5. Objective test results (G.723.1 at 5.3 kbit/s)

Method	Average PESQ		
	Male	Female	Average
Tandem A to B	3.352	3.071	3.212
Transcoding A to B	3.357	3.072	3.215
Tandem B to A	3.389	3.099	3.244
Transcoding B to A	3.378	3.128	3.253

Notice: A to B (from G.729 to G.723.1), B to A (from G.723.1 to G.729).

Table 6. Objective test results (G.723.1 at 6.3 kbit/s)

Method	Average PESQ		
	Male	Female	Average
Tandem A to B	3.459	3.151	3.305
Transcoding A to B	3.496	3.171	3.334
Tandem B to A	3.513	3.211	3.362
Transcoding B to A	3.471	3.215	3.343

Notice: A to B (from G.729 to G.723.1), B to A (from G.723.1 to G.729).

Table 7. Preference test results

Method	Average MOS			
	G.723.1 at 5.3 kbit/s		G.723.1 at 6.3 kbit/s	
from G.729 to G.723.1	Male	Female	Male	Female
Transcoding	4.12	3.92	4.51	4.28
Tandem	3.98	3.75	4.38	4.16

Table 8. Preference test results

Method	Average MOS			
	G.723.1 at 5.3 kbit/s		G.723.1 at 6.3 kbit/s	
from G.723.1 to G.729	Male	Female	Male	Female
Transcoding	4.19	3.98	4.25	4.21
Tandem	4.23	3.96	4.27	4.09

the G.723.1 (G.729) encoder are not required. Therefore, computations are significantly reduced while maintaining good speech quality with no additional look-ahead delay. In addition, as mentioned earlier in Section 3.1.5 and Section 3.2.3, we propose two fast FCB search algorithms to reduce computation of ACELP coding for G.723.1 and for G.729. We propose two fast search algorithms to reduce computation of the ACB and MP-MLQ coding for G.723.1 as mentioned earlier in Section 3.1.3 and 3.1.4, respectively.

We analyzed the distribution of computational load of the tandem approach and the proposed transcoding method and the results are shown in Table 9. In the G.723.1 6.3 kbit/s encoder, the proposed fast search algorithms can reduce the computational complexity by about $23\% \times 0.815 \approx 18.75\%$ and $55\% \times 0.9599 \approx 52.8\%$ relative to the original ACB and MP-MLQ search procedures, respectively. For G.723.1 5.3kbit/s encoder, the proposed method can reduce the computational complexity by about $27\% \times 0.815 \approx 22\%$ and $47\% \times 0.9009 \approx 42.35\%$ compared with the original ACB and ACELP search procedures, respectively. Overall, the proposed fast search algorithms reduce computational complexity by about 84.55% and 80.35% compared with the original G.723.1 6.3 and 5.3 kbit/s, respectively.

Similarly, in the G.729 encoder, the proposed fast FCB search algorithm can reduce the computational complexity by $21\% \times 0.9158 \approx 19.23\%$ compared with the original ACELP search procedure. In other words, the proposed method searched the FCB that required 1.77 % $(21\% - 19.23\%)$ computations. The other values in Table 9 were obtained through calculations similar to those for the 1.77% result. For the calculation of the value 0.9158, please refer to Section 3.2.3. Overall, the proposed method can reduce computational complexity by 49.23% compared with the original G.729 search.

Finally, we proposed fast search algorithms that can reduce the average computational complexity by about 82.45% and 49.23% compared with the original G.723.1 and G.729 codecs search procedure, respectively. Therefore, the proposed transcoding algorithm can improve the overall computational complexity by about 65.8% compared with the tandem approach.

4.4. **Transmission delay evaluation.** The total delay of a speech communication system is mainly comprised of three types of delay: processing delay, algorithmic delay and network delay. This paper does not consider network delay. For transmission from G.729 to G.723.1, the total delays of the tandem method and the proposed transcoding method are denoted as

$$D_{A->B}^{\text{tandem}} = 42.5 + 3 \times (D_A^{En} + D_A^{De}) + D_B^{En} + D_B^{De} \tag{15}$$

$$D_{A->B}^{\text{transcoding}} = 35 + 3 \times (D_A^{En} + D_A^{De}) + D_{A->B}^{Tr} + D_B^{De} \tag{16}$$

where subscripts A and B denote G.729 and G.723.1 codecs, respectively, and $A->B$ and $B->A$ denote the conversion from G.729 to G.723.1 and from G.723.1 to G.729, respectively. In addition, the superscripts En, De and Tr denote encoding, decoding and transcoding, respectively. The transcoding delay, $D_{A->B}^{Tr}$, includes LSP, open-loop pitch conversion processing time, and the algorithmic delay of the proposed method in the G.723.1 codec. D_B^{En} and D_B^{De} denote the delays produced by encoding and decoding of the G.723.1 codec, respectively. To obtain translation parameters between the G.729 and G.723.1 speech coders, the frame size needs to be synchronized. For this purpose, we take the least common multiple of the frame size of the two coders. Figure 6 shows the relationship between frames and subframes used in the G.729 and G.723.1 coders. Here, the least common multiple of the frame size is three, as the frame size of the G.723.1 codec (30 ms) is three times the frame size of the G.729 codec (10 ms). For the tandem type, the decode-then-encode approach can be considered to achieve direct communication between G.729 to G.723.1 speech codec. The process can be described intuitively as follows: We start with G.729 encoding three frames (30 ms, $3 \times D_A^{En}$) and then the decoding process reconstructs the compressed speech ($3 \times D_A^{De}$) and then perform G.723.1 encoding to complete the transcoding. It should be noted that a 7.5 ms look-ahead time is needed for LPC analysis of G.723.1 [4]. Similarly, a 5 ms look-ahead time that is needed for

Table 9. The distribution of computational load of the tandem approach and the proposed transcoding method

Function description	G.723.1, 6.3 kbps		G.723.1, 5.3 kbps		G.729, 8 kbps	
	Tandem	Transcoding	Tandem	Transcoding	Tandem	Transcoding
LPC	1%	–*	2%	–*	8%	–*
Open-loop pitch	5%	–*	6%	–*	10%	–*
LSP	7%	–*	8%	–*	12%	–*
Filtering	9%	9%	10%	10%	35%	35%
ACB	23%	4.25%	27%	5%	14%	14%
FCB	55%	2.2%	47%	4.65%	21%	1.77%
Total	100%	15.45%	100%	19.65%	100%	50.77%
save the computational complexity	84.55%		80.35%		49.23%	

–* The function process can be omitted.

Table 10. Transmission delay (ms) (from G.729 to G.723.1)

Operation	Coder	Tandem	Coder	Proposed
Buffering	G.729	35	G.729	35
Encoding		$3 \times D_A^{En}$		$3 \times D_A^{En}$
Decoding		$3 \times D_A^{De}$		$3 \times D_A^{De}$
Encoding	G.723.1	D_B^{En}	Transcoding	$D_{A->B}^{Tr}$
Delay		7.5		–*
Decoding		D_B^{De}	G.723.1	D_B^{De}
Total Delay	$42.5 + 3 \times (D_A^{En} + D_A^{De}) + D_B^{En} + D_B^{De}$		$35 + 3 \times (D_A^{En} + D_A^{De}) + D_{A->B}^{Tr} + D_B^{De}$	

–* The LPC analysis of the G.723.1 can be omitted. Three frames (30 ms) + 5 ms look-ahead time = 35 ms.

Table 11. Transmission delay (ms) (from G.723.1 to G.729)

Operation	Coder	Tandem	Coder	Proposed
Buffering	G.723.1	37.5	G.723.1	37.5
Encoding		D_B^{En}		D_B^{En}
Decoding		D_B^{De}		D_B^{De}
Encoding	G.729	$3 \times D_A^{En}$	Transcoding	$D_{B->A}^{Tr}$
Delay		5		–*
Decoding		$3 \times D_A^{De}$	G.729	$3 \times D_A^{De}$
Total Delay	$42.5 + 3 \times (D_A^{En} + D_A^{De}) + D_B^{En} + D_B^{De}$		$37.5 + D_B^{En} + D_B^{De} + D_{B->A}^{Tr} + 3 \times D_A^{De}$	

–* The LPC analysis of the G.729 can be omitted. One frame (30 ms) + 7.5 ms look-ahead time = 37.5 ms.

LPC analysis of G.729 [10]. As a result, the G.729 codec has a total algorithmic delay of 35 ms. However, the proposed LSP conversion scheme does not introduce this look-ahead delay because the processing of LPC analysis is not executed. Table 10 shows that the transcoding delay of the $D_{A->B}^{Tr}$ is much less than that of D_B^{En} and thus, the total delay will be further reduced. The transmission delays of the tandem approach and the proposed transcoding scheme are summarized in Table 10.

Similarly, for conversion from G.723.1 to G.729 under the tandem approach, the G.723.1 codec has a total algorithmic delay of 37.5 ms. The total delays of the tandem method and the proposed transcoding method, denoted by $D_{B->A}^{\text{tandem}}$ and $D_{B->A}^{\text{transcoding}}$, respectively, and transmission delays are summarized in Table 11.

$$D_{B->A}^{\text{tandem}} = 42.5 + 3 \times (D_A^{En} + D_A^{De}) + D_B^{En} + D_B^{De} \tag{17}$$

$$D_{B->A}^{\text{transcoding}} = 37.5 + D_B^{En} + D_B^{De} + D_{B\to A}^{Tr} + 3 \times D_A^{De} \tag{18}$$

5. **Conclusion.** In this paper, we proposed an efficient transcoding scheme that is able to convert 5.3 and 6.3 kbit/s G.723.1 parameters into a 8 kbit/s G.729 parameters. The proposed transcoding scheme is comprised of four processes: LSP conversion, open-loop pitch conversion, fast adaptive-codebook search, and fast fixed-codebook search. Using the proposed method, the stochastic excitation search loops and search pitch gain-vectors can be reduced; open-loop pitch prediction and LPC coefficients computation can be omitted in the G.729 (G.723.1) encoder. Therefore, the proposed transcoding scheme can improve upon several inherent disadvantages of the tandem approach, such as speech quality degradation, high computational complexity, and longer coding delay time. Subjective and objective evaluation results showed that the proposed transcoding algorithm produces speech quality similar to the tandem approach with shorter delays and less computational complexity.

Acknowledgment. This work was supported by the National Science Council of Taiwan under research contract NSC- 100-2221-E-218 -052. The authors would like to thank the reviewers for their valuable comments and suggestions.

REFERENCES

[1] G. Thomsen and Y. Jani, Internet telephony: Going like crazy, *IEEE Spectrum*, vol.37, no.5, pp.52-58, 2000.

[2] X. Wang, J. Lin, Y. Sun, H. Gan and L. Yao, Applying feature extraction of speech recognition on VoIP auditing, *International Journal of Innovative Computing, Information and Control*, vol.5, no.7, pp.1851-1856, 2009.

[3] J. Dong, X. Wei, Q. Zhang and L. Zhao, Speech enhancement algorithm based on higher-order cumulates parameter estimation, *International Journal of Innovative Computing, Information and Control*, vol.5, no.9, pp.2725-2733, 2009.

[4] ITU-T Rec. G.723.1, *Dual Rate Speech Coder for Multimedia Communications at 5.3 and 6.3 kbit/s*, 1996.

[5] J. F. Yang, R. S. Lin and S. Hu, Time-varied pitch gain predictor for low bit rate speech coders, *IEICE Trans. Information and Systems*, vol.E85-D, no.4, pp.751-758, 2002.

[6] S. K. Jung, K. T. Kim, Y. C. Park and H. G. Kang, A fast adaptive-codebook search algorithm for G.723.1 speech coder, *IEEE Signal Processing Letters*, vol.12, no.1, pp.75-78, 2005.

[7] F. K. Chen, J. F. Yang and Y. L. Yan, Candidate scheme for fast ACELP search, *IEE Proc. of Vision, Image, and Signal Processing*, vol.149, no.1, pp.10-16, 2002.

[8] C. Negrescu, Optimization algorithm for the MP-MLQ excitation in G723.1 encoder, *IEEE Conf. Electronics, Circuits and Systems*, pp.1003-1006, 2006.

[9] S. M. Tsai and J. F. Yang, Efficient algebraic code-excited linear predictive codebook search, *IEE Proc. of Vision, Image, and Signal Processing,* vol.153, no.61, pp.761-768, 2006.

[10] ITU-T Rec.G.729, *Coding of Speech at 8 kbit/s using Conjugate Structure Algebraic Code Excited Linear Prediction*, 1996.

[11] ITU-T Rec. G.729-Annex A, *Reduced Complexity 8 kbit/s CS-ACELP Speech Codec*, 1996.

[12] ITU-T Rec. H.324, *Terminal for Low Bit Rate Multimedia Communication*, 1996.

[13] ITU-T Rec. H.323, *Visual Telephone Systems and Equipment for Local Area Networks Which Provide a Nonguaranteed Quality of Service*, 1996.

[14] S. F. C. Neto and F. L. Crcoran, Performance assessment of tandem connection of enhanced cellular coders, *IEEE Proc. of International Conference on Acoustics Speech Signal Processing*, pp.177-180, 1999.

[15] H. G. Kang, H. K. Kim and R. V. Cox, Improving transcoding capability of speech coders in clean and frame Erasured channel environments, *Proc. of IEEE Workshop on Speech Coding*, pp.78-80, 2000.

[16] S. M. Tsai and J. F. Yang, GSM to G.729 speech transcoder, *The 8th IEEE International Conference on Electronics, Circuits and Systems*, vol.1, pp.485-488, 2001.

[17] S. Ruslan, H. Ludmila and B. Pavlo, Method of converting speech codec formats between GSM 06.20 and G.729, *IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, vol.6, no.8, pp.686-689, 2007.

[18] S. W. Yoon, S. K. Jung, Y. C. Park and D. H. Youn, An efficient transcoding algorithm for G.723.1 and G.729A speech coders, *Proc. of Eurospeech*, pp.2499-2502, 2001.

[19] S. Ruslan, Method of converting speech codec formats between G.723.1 and G.729, *CADSM'2007*, pp.483-486, 2007.

[20] S. W. Yoon, H. G. Kang, Y. C. Park and D. H. Youn, An efficient transcoding algorithm for G.723.1 and G.729A speech coders: Interoperability between mobile and IP network, *Speech Communication*, vol.43, no.1-2, pp.17-31, 2004.

[21] 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.

[22] R. S. Lin, Y. C. Chen and F. K. Chen, Low complexity search method for G.723.1 MP-MLQ algorithm, *The 8th International Conference on Intelligent Systems Design and Applications*, vol.1, pp.158-161, 2008.

[23] S. M. Lee, S. Park and Y. Jang, Cost-effective implementation of ITU-T G.723.1 on A DSP chip, *Proc. of 1997 IEEE International Symposium on Consumer Electronics*, pp.31-34, 1997.

[24] 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*, 2001.

[25] W. C. Chu, *Speech Coding Algorithms: Foundation and Evolution of Standardized Coders*, John Wiley & Sons, Inc., 2003.

[26] *3GPP TS 26.090: ARM Speech Codec; Speech Transcoding Function, 3GPP*, 2007.