RESEARCH   **Open Access**

An efficient algebraic codebook search for ACELP speech coder

Ning-Yun Ku[1], Cheng-Yu Yeh[2*] and Shaw-Hwa Hwang[1]

Abstract

In a bid to enhance the search performance, this paper presents an improved version of reduced candidate mechanism (RCM), an algebraic codebook search conducted on an algebraic code-excited linear prediction (ACELP) speech coder. This improvement is made based on two findings in a piece of our prior work. The first finding is that a pulse with a high contribution in the associated track is more likely to serve as an optimal pulse in the optimal codevector and the second is that the speech quality can be well maintained at a search accuracy above 50% approximately. Subsequently, a new finding in this study concerning a structured algebraic codebook in G.729 indicates that there is a 0.8321 probability that the number 1 ranked pulse in a global sorting by pulse contribution is indeed one of the optimal pulses. Hence, the number 1 pulse in the global sorting is labeled as one of the optimal pulses, following which a sequence of search tasks are fulfilled through RCM. This proposed complexity reduction algorithm, implemented on a G.729A speech codec, takes as few as eight searches, a search load tantamount to 2.5% of G.729A, 12.5% of global pulse replacement method (iteration = 2), 16.7% of iteration-free pulse replacement method, and 50% of RCM ($N = 2$). This proposal is thus found to successfully reduce the required computational complexity to a great extent as intended.

Keywords: Speech codec; VoIP; Algebraic code-excited linear-prediction (ACELP); Algebraic codebook search; Reduced candidate mechanism (RCM)

1 Introduction

An algebraic code-excited linear prediction (ACELP)-based speech coding technique [1-3] is a type of technique most widely applied to digital speech communication systems and serves as a mainstream technique adopted in a great number of speech coding standards due to a double advantage of low bit rates and high speech quality. The main coding flow for an ACELP coding technique is to perform a linear predictive coding (LPC) on an input speech signal and then perform an adaptive as well as an algebraic codebook search on an LPC residual signal.

Yet, the price paid is a high computational complexity requirement, particularly in an algebraic codebook search, due to the reason that it necessitates a tremendous computational load when conducting a full search over the algebraic codebook to locate the optimal pulses. As suggested in [4], the computational load required in

a G.729 algebraic codebook search is dominated by two parts, namely, the load in a search process and the load during the algorithm initialization phase. The former and the latter respectively account for 74.9% and the remaining 25.1% of the entire computational load. Provided that there is a way to reduce the computational load to a great extent, an ACELP-based coding technique can be extensively applied to an embedded system on a handheld device. In this way, a high-performance embedded system is not seen as required, making electronic devices cost-competitive. Moreover, due to a computational load reduction, the aim of complexity saving is reached for an extended operation time period.

For this sake, full search scheme is hardly adopted in most prominent speech coding standards. There have been a great number of studies proposed on search load reduction, say, the focus search in G.729 [3] and the depth-first search in G.729A [5], among other approaches. In recent times, to further reduce the computational complexity, there are many effective algebraic codebook search algorithms [6-14] proposed to reduce

* Correspondence: cy.yeh@ncut.edu.tw
[2]Department of Electrical Engineering, National Chin-Yi University of Technology, 57, Sec. 2, Zhongshan Rd., Taiping Dist, Taichung 41170, Taiwan
Full list of author information is available at the end of the article

the pulse combinations. For example, the candidate scheme [6] attempts to decrease the number of the candidate positions. The least important pulse replacement [7] is presented to repeatedly measure the contribution of each pulse and replace the least important pulse with a new one. Analogous to [7], the global pulse replacement (GPR) [8], adopted by G.729.1 [9], and the iteration-free pulse replacement (IFPR) [10,11] methods are proposed to further efficiently reduce the search complexity.

Alternatively, the reduced candidate mechanism (RCM) approach [12], a piece of our prior work, has been presented. In RCM, individual pulse contribution is evaluated in the associated track and sorted in descending order. Subsequently, a full search is performed on the sorted top N pulses treated as candidates. In this way, the optimal pulse combination is acquired following N^4 searches, that is, a significant reduction in search complexity.

It is a significant finding in RCM that a pulse with a high contribution in the associated track is more likely to serve as the optimal pulse in the optimal codevector. Thus, a further investigation is conducted into the RCM approach, and an improved version of RCM is proposed for further load reduction while comparable speech quality is well maintained. In contrast to existing approaches, this proposal, implemented in a G.729A speech codec, is presented as a superior choice in the aspect of search complexity and as a competitive candidate in terms of speech quality.

The rest of the paper is organized as follows. The coding criterion of an algebraic codebook in G.729 and various search methods are briefly reviewed in Section 2. Section 3 proposes an efficient approach for the purpose of search complexity reduction. Experimental results are demonstrated and discussed in Section 4. This work is concluded at the end of this paper.

2 Algebraic codebook search

With the determination of an optimal codevector as the goal of the algebraic codebook search, the codebook in G.729 is configured as tabulated in Table 1, on the basis of which each codevector contains four nonzero pulses extracted out of associated track. Each pulse's amplitude can be either +1 or −1.

Table 1 A structured algebraic codebook in G.729

Track	Pulse	Sign	Positions
T_0	i_0	s_0	m_0: 0, 5, 10, 15, 20, 25, 30, 35
T_1	i_1	s_1	m_1: 1, 6, 11, 16, 21, 26, 31, 36
T_2	i_2	s_2	m_2: 2, 7, 12, 17, 22, 27, 32, 37
T_3	i_3	s_3	m_3: 3, 8, 13, 18, 23, 28, 33, 38
			4, 9, 14, 19, 24, 29, 34, 39

The optimal codevector $\mathbf{c}_k = \{c_k(n)\}$ is thus found by minimizing the mean squared weighted error between the original and the synthesized speeches [2,3], defined as

$$\varepsilon_k = \|\mathbf{x} - g\mathbf{H}\mathbf{c}_k\|^2 \tag{1}$$

where $\mathbf{x}$ denotes the target vector, g a scaling gain factor, and $\mathbf{H}$ a lower triangular convolution matrix. It can be shown that the optimal codevector is the one maximizing the term Q_k:

$$Q_k = \frac{(\mathbf{x}^{\mathrm{T}}\mathbf{H}\mathbf{c}_k)^2}{\mathbf{c}_k^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{c}_k} = \frac{(\mathbf{d}\mathbf{c}_k)^2}{\mathbf{c}_k^{\mathrm{T}}\mathbf{\Phi}\mathbf{c}_k} \tag{2}$$

where $\mathbf{d} = \mathbf{x}^{\mathrm{T}}\mathbf{H}$, the correlation function, is expressed as

$$d(n) = \sum_{i=n}^{L-1} x(i)h(i-n), \quad 0 \le n \le L-1 \tag{3}$$
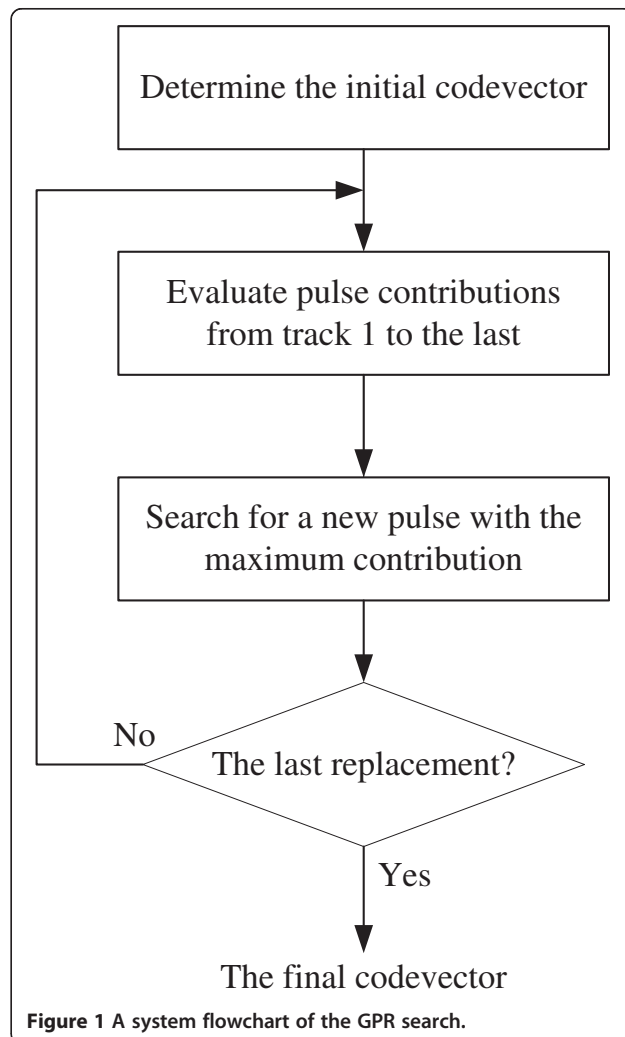

Figure 1 A system flowchart of the GPR search.

where L is the speech subframe size. The correlations of $h(n)$, are contained in the symmetric matrix $\mathbf{\Phi} = \mathbf{H}^{\mathrm{T}}\mathbf{H}$, where the entries are given by

$$\varphi(i,j) = \sum_{n=j}^{L-1} h(n-i)h(n-j), \quad 0 \leq i \leq L-1; \quad i \leq j \leq L-1$$

(4)

It takes a total of 8,192 ($8 \times 8 \times 8 \times 16$) searches, a tremendous computational load, to conduct a full search, i.e., repeated computations and comparisons in (2), for the identification of the optimal codevector. Therefore, a focused search method is adopted in G.729 to reduce the search times to below 1,440. However, the search number is further reduced to 320, adopting a depth-first tree search method in G.729A. Besides, three existing methods, the GPR, IFPR, and RCM methods, will be discussed in this section.

2.1 The GPR search method
The GPR method stems from the least important pulse replacement algorithm [7]. In order to prevent the termination of the pulse replacement procedure without finding the optimal codevector in the GPR algorithm, except for the only track that contains the least important pulse, all the tracks are searched for a new pulse. That is, the new pulse is sought by replacing each pulse in each track with a new one so that the Q_k associated with a new codevector is maximized. On the ground that the variation in Q_k is always maximized during the replacement procedure, the codevector approaches the optimal solution rapidly as this procedure is repeated. When the value of Q_k once reaches the upper bound, the search procedure is then terminated.

A system flowchart of the GPR method is sketched in Figure 1. Following an application of the GPR method to G.729A at the first stage, the initial Q_k is evaluated and
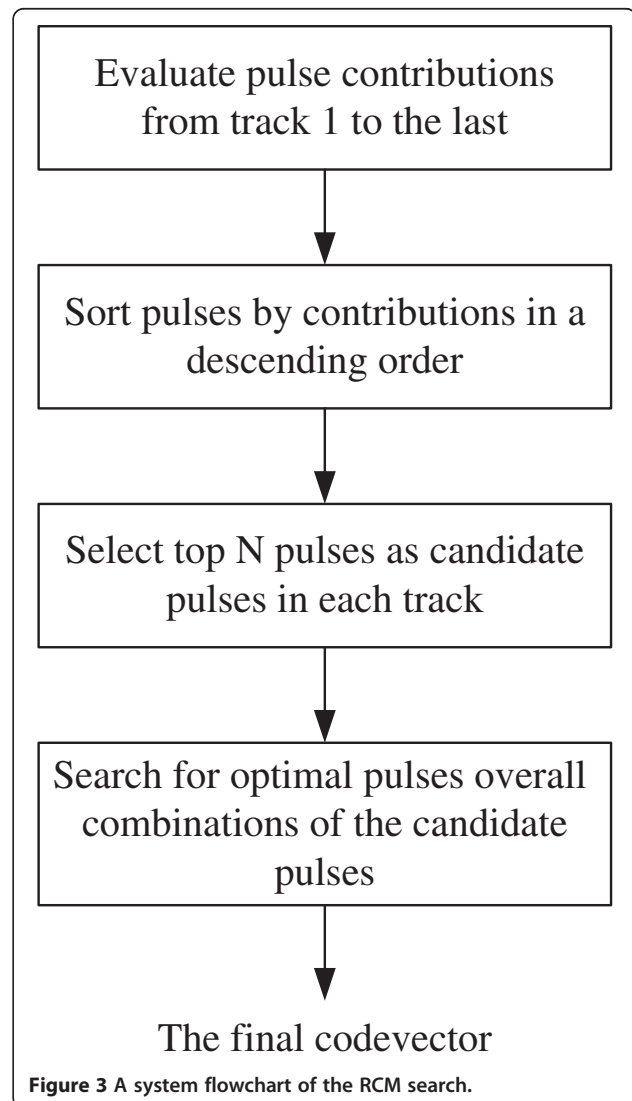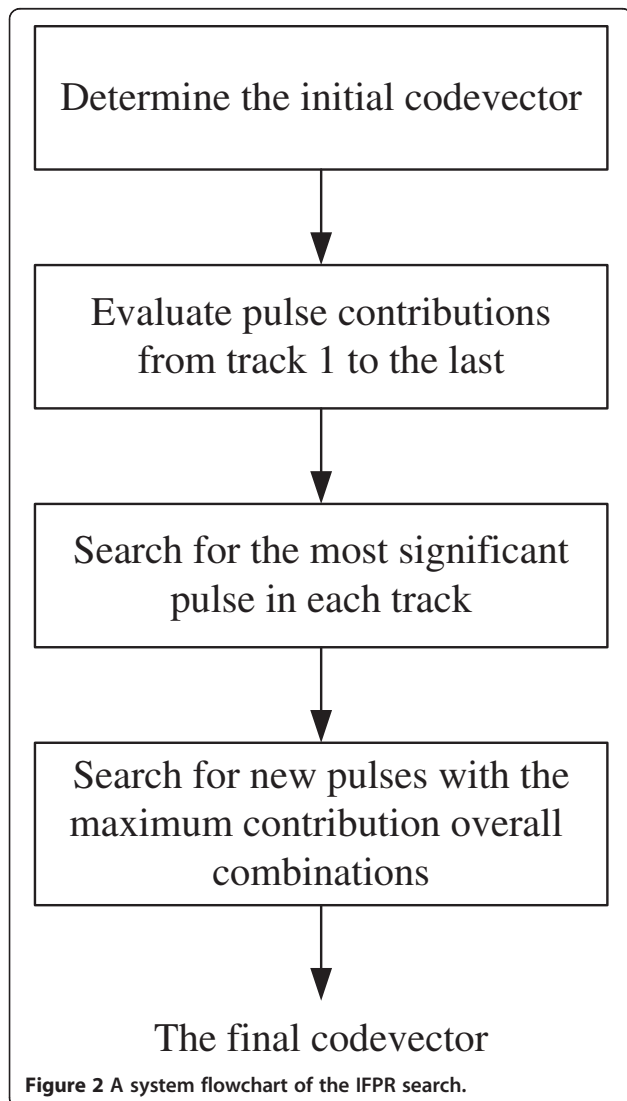

Figure 2 A system flowchart of the IFPR search.

Figure 3 A system flowchart of the RCM search.

Table 2 Hit probability of each pulse contribution in a global sorting

Global sorting for pulse contribution	Hit probability	Global sorting for pulse contribution	Hit probability
1	0.8321	21	0.0189
2	0.5857	22	0.0161
3	0.4589	23	0.0144
4	0.3621	24	0.0127
5	0.2925	25	0.0116
6	0.2383	26	0.0103
7	0.1954	27	0.0096
8	0.1595	28	0.0084
9	0.1314	29	0.0079
10	0.1085	30	0.0071
11	0.0908	31	0.0063
12	0.0760	32	0.0059
13	0.0635	33	0.0054
14	0.0527	34	0.0048
15	0.0450	35	0.0047
16	0.0385	36	0.0042
17	0.0329	37	0.0040
18	0.0281	38	0.0037
19	0.0245	39	0.0032
20	0.0213	40	0.0030

the initial codevector is yielded with a single search. At the second stage, it requires 36 searches to seek the new pulse during the first pulse replacement procedure and requires an average of 27 during the second. Therefore, the overall search complexity is evaluated as $37 + 27 \times (R - 1)$ for $R \geq 1$, where R is the number of iterations of the pulse replacement procedure.

2.2 The IFPR search method

In the previously mentioned pulse replacement methods, [7] and [8], the computational load increases with the number of iterations of the replacement procedure. In the IFPR method, new pulses are sought by a number of pulse replacements at a time following pulse contributions evaluated for every track so as to maximize over all combinations a search criterion, which replaces the pulses pertaining to the initial codevector with the most significant pulses for every track.

Presented in Figure 2 is a system flowchart of the IFPR method. Applying IFPR method to G.729A at the first stage, the initial Q_k is evaluated and the initial codevector is then yielded with a single search. At the second stage, a total of 36 searches are performed to measure the pulse contribution so as to sort out the most significant pulses in each track. In order to find the final codevector in the end, it requires a total of 11 searches for all combinations, i.e., the number required from two pulses replacement to four pulses replacement, to replace the pulses of the initial codevector with the most significant pulses for every track, that is, an overall search complexity of 48.

2.3 The RCM search method

Ahead of a search task, the number of candidate pulses in each track is reduced for the purpose of search complexity reduction. This is done in this work according to the contribution of individual pulses. It is that in each track, a pulse sorting is made by the contribution thereof in descending order as the first step, and then, the top N pulses are chosen as the candidate pulses for a full search. In this way, the search process needs to be performed for merely N^4 number of times for the optimal pulse combination, and the search complexity is reduced remarkably in particular for low values of N.

The contribution made by individual pulses is given as (2), that is, a higher value of Q_k reflects a higher
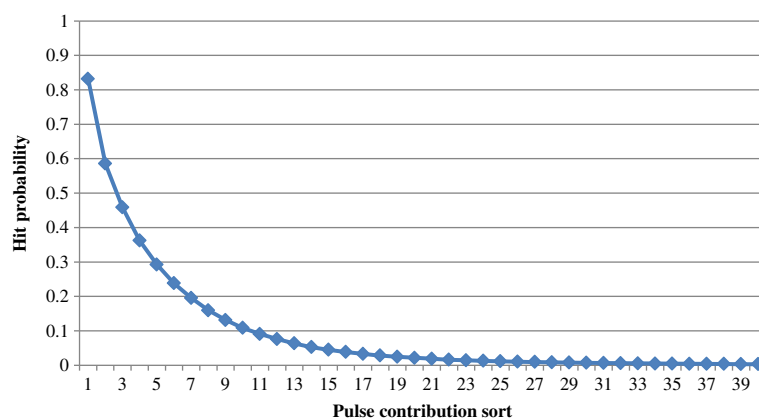

Figure 4 Plot of the hit probability of each pulse contribution in a global sorting.

contribution. In consideration of merely a single pulse contribution, the number of nonzero pulses in the codevector $\mathbf{c}_k$ of length 40 is reduced to 1 from 4. Therefore, (2) can be simplified into (5), where the numerator of (5) is derived from (2) and (3), and the denominator of (5) is derived from (2) and (4), respectively. Just as in (2), the contribution of the ith pulse is reflected by the value of Q_k^i

$$Q_k^i = \frac{d^2(i)}{\varphi(i,i)}, \quad 0 \le i \le L-1 \tag{5}$$

In [12], it is verified that a single pulse with a higher contribution within each track is more likely to be the optimal pulse out of the optimal codevector within the associated track. Thus, the RCM approach is used to reduce the search complexity by the reduction in the number of candidate pulses in each track. This approach is decomposed into two stages as follows. The first is to evaluate individual pulse contribution with (5), which indicates that a higher value of Q_k^i denotes a higher pulse contribution. Subsequently, the top N pulses, $1 \le N \le 8$, are extracted out of the Q_k^i sorting in each track as the prerequisite of the second stage. Then, in the second stage, it is proven that the pulses' combination with the highest value of Q_k, as given in (2), is indeed the optimal solution through a nest-loop search. A flowchart of the RCM approach is presented in Figure 3. The first three steps in Figure 3 are to compute the contribution of each pulse in each track and then to select the candidate pulses in each track. The last step is to compute the best combination of the candidates.

3 Proposed approach
In the aspect of RCM, the first significant finding indicates that a pulse with a high contribution is more likely to serve as one of the optimal pulses in the associated track, whereby the hit probability can be elevated when conducting a search task. The second finding reveals that the speech quality can be well maintained as long as the search accuracy exceeds a threshold, say 50% in [12]. Based on such findings, an improved version of RCM, referred to as the Fixed-G1-RCM approach, is presented to achieve the aim of search load improvement in the absence of speech quality degradation. As its name indicates, the top 1 pulse contribution in a global sorting, termed as the G1 pulse, is presumed to be one of the four optimal pulses, following which the rest of optimal pulses are located over the remaining three tracks through RCM.

Thus, it is an issue of our interest whether there exists a high correlation between the top 1 pulse contribution and the possibility that such pulse is indeed one of the

optimal pulses. Hence, over entire tracks, a hit probability $p_h(n)$ in a global sorting is defined as

$$p_h(n) = \frac{\mathrm{NH}(n)}{\mathrm{TSF}}, \quad 1 \le n \le 40 \tag{6}$$

where $\mathrm{NH}(n)$ denotes the number of times that the nth pulse, in terms of the contribution priority, hits the optimal codevector and TSF the total number of testing subframes.
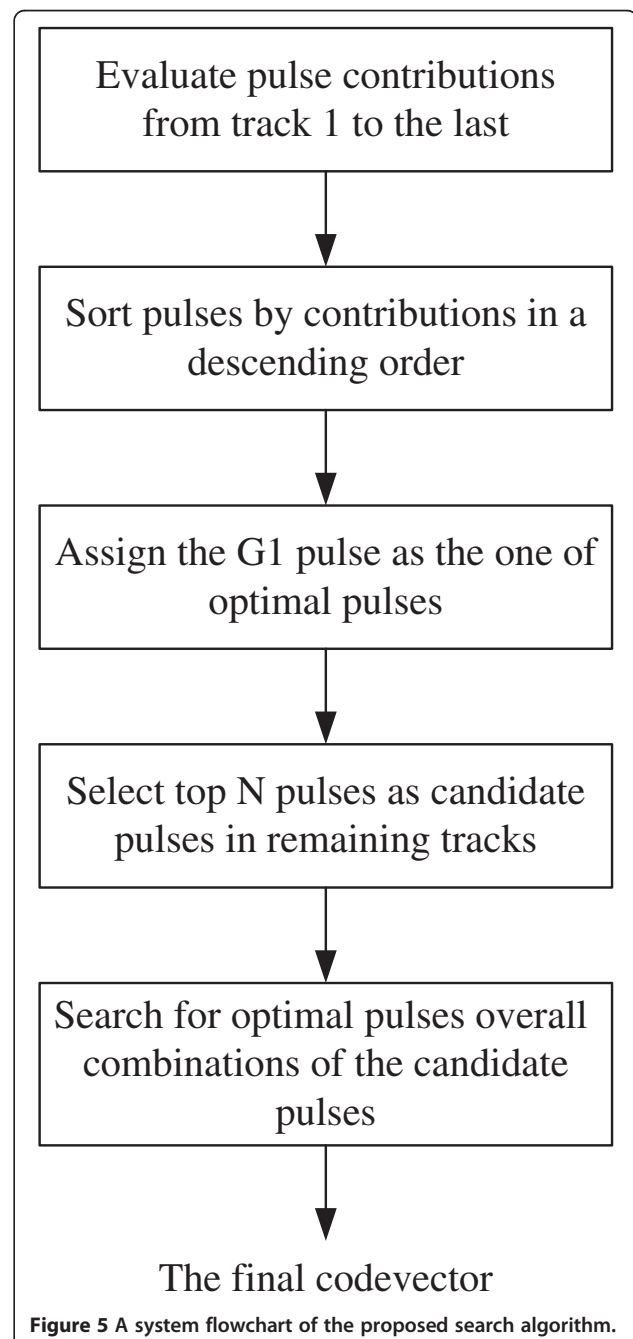

Evaluate pulse contributions from track 1 to the last

Sort pulses by contributions in a descending order

Assign the G1 pulse as the one of optimal pulses

Select top N pulses as candidate pulses in remaining tracks

Search for optimal pulses overall combinations of the candidate pulses

The final codevector

Figure 5 A system flowchart of the proposed search algorithm.

Table 3 Search accuracy comparison among various methods

Methods		Search accuracy for locating various numbers of intended pulses (%)				
		1 pulse	2 pulses	3 pulses	4 pulses (all right)	0 pulse (all wrong)
G.729A		98.3475	92.1456	80.9918	68.6438	1.6525
GPR	$R = 1$	98.7032	90.7750	76.1053	55.0718	1.2968
	$R = 2$	98.4946	91.8335	80.4779	76.1053	1.5054
	$R = 3$	98.5246	92.1081	81.2227	80.4779	1.4754
	$R = 4$	98.5283	92.1547	81.3393	81.2227	1.4717
IFPR		98.6810	92.4963	80.0048	68.0824	1.3190
RCM	$N = 1$	99.3295	89.0873	55.8720	17.3353	0.6705
	$N = 2$	98.3009	90.8486	73.6532	50.3579	1.6991
	$N = 3$	98.8545	94.0394	83.5763	70.7329	1.1455
	$N = 4$	99.2014	95.9617	89.2187	81.8716	0.7986
Proposed method	$N = 1$	99.3295	89.0873	55.8720	17.3353	0.6705
	$N = 2$	98.2054	89.8500	70.0755	44.0814	1.7946
	$N = 3$	98.2060	91.2376	76.7365	60.2498	1.7940
	$N = 4$	98.2369	92.1541	80.4684	69.0717	1.7631

Subsequently, a global sorting is conducted by pulse contribution over entire tracks. As tabulated in Table 2, it is seen in the G.729 algebraic codebook that there is a 0.8321 hit probability that the number 1 pulse is indeed one of the optimal pulses, while the hit probability drops dramatically from 0.8321 to 0.5857 in case the number 2 pulse acts as the optimal one. A graphic illustration of Table 2 is shown in Figure 4. The statistics on Table 2 is based on a speech database in Chinese language, containing 9,650 syllables out of 100 sentences for a duration over 41 min and 495,608 subframes, that is, TSF = 495,608.

The above analysis confirms that the top 1 pulse contribution in a global sorting has a highly hit probability that such pulse is indeed the one of optimal pulses. Thus, the Fixed-G1-RCM approach is presented in this study as an efficient way to further speed up the searching process. In this approach, the G1 pulse is presumed to be one of the four optimal pulses, following which the rest of optimal pulses are located over the remaining three tracks through RCM. In this context, the number of searches required is reduced to N^3 for $N \geq 2$. This proposal, as opposed to RCM, is developed in an attempt to considerably reduce the search load to N^3 from N^4 for identical value of N. Furthermore, since there is a high probability, up to 0.8321, that the top 1 pulse in the global sorting hits one of the optimal pulses, the speech quality can be well maintained on a condition that a search accuracy of approximately 50% is reached. Finally, this proposal is implemented as follows and is shown in Figure 5.

Step 1 Individual pulse contribution is evaluated by (5), and a sorting is made by pulse contribution within the associated track.

Step 2 The one with the global maximum pulse contribution, named as G1, is located out of all the top 1 pulses among all the tracks.

Step 3 G1 is presumed to be one of four optimal pulses, and then, the value of N is determined for the searching task conducted over the remaining three tracks through RCM.

Step 4 The searching task terminates the moment the combination of optimal pulses is acquired.

Furthermore, this proposal can be applied to an algebraic codebook search for other ACELP-based coders. Assuming that there are Nt tracks in a structured algebraic

Table 4 Search load comparison among various methods

Methods		Search load
G.729A		320
GPR	$R = 1$	37
	$R = 2$	64
	$R = 3$	91
	$R = 4$	118
IFPR		48
RCM	$N = 1$	1
	$N = 2$	16
	$N = 3$	81
	$N = 4$	256
Proposed method	$N = 1$	1
	$N = 2$	8
	$N = 3$	27
	$N = 4$	64

Table 5 Objective speech quality comparison among various methods

Methods		PESQ				MOS-LQO			
		Mean	STD	Max.	Min.	Mean	STD	Max.	Min.
G.729A		3.8126	0.0838	3.9640	3.5150	3.9502	0.1026	4.1230	3.5670
GPR	$R = 1$	3.8145	0.0757	4.0850	3.5640	3.9521	0.0883	4.2350	3.6410
	$R = 2$	3.8324	0.0777	4.0490	3.5420	3.9725	0.0906	4.2010	3.6110
	$R = 3$	3.8376	0.0739	3.9830	3.5950	3.9786	0.0859	4.1370	3.6830
	$R = 4$	3.8382	0.0749	4.0350	3.5560	3.9793	0.0867	4.1880	3.6310
IFPR		3.8276	0.0761	4.0830	3.5820	3.9672	0.0884	4.2330	3.6650
RCM	$N = 1$	3.7083	0.0807	3.9050	3.3230	3.8250	0.1026	4.0550	3.3030
	$N = 2$	3.8064	0.0819	4.0770	3.5190	3.9423	0.0971	4.2280	3.5800
	$N = 3$	3.8256	0.0740	3.9880	3.5850	3.9651	0.0868	4.1420	3.6700
	$N = 4$	3.8331	0.0735	3.9710	3.5570	3.9735	0.0859	4.1240	3.6320
Proposed method	$N = 1$	3.7083	0.0807	3.9050	3.3230	3.8250	0.1026	4.0550	3.3030
	$N = 2$	3.7937	0.0775	4.0220	3.5120	3.9279	0.0927	4.1760	3.5700
	$N = 3$	3.8143	0.0766	4.0270	3.5640	3.9518	0.0906	4.1800	3.6410
	$N = 4$	3.8312	0.0682	4.0320	3.6100	3.9717	0.0791	4.1860	3.7020

codebook, out of which Np pulses must be selected in each codebook search, then the respective numbers of searches required in RCM and this presented Fixed-G1-RCM searching approaches are given as

$$\text{Searches}(\text{RCM}, N) = \left(C_{Np}^{N}\right)^{Nt} = \left(\frac{N!}{Np!(N-Np)!}\right)^{Nt} \tag{7}$$

$$\text{Searches}(\text{Fixed-G1-RCM}, N) = C_{Np-1}^{N-1}\left(C_{Np}^{N}\right)^{(Nt-1)}$$
$$= \frac{(N-1)!}{(Np-1)!(N-Np)!}\left(\frac{N!}{Np!(N-Np)!}\right)^{(Nt-1)} \tag{8}$$

Taking a structured algebraic codebook in G.729 as an instance ($Nt = 4$, $Np = 1$), this proposal requires N^3 searches, while RCM requires as many as N^4 searches. In an arbitrary structured algebraic codebook, there is no doubt in this study, as opposed to RCM, that the G1 pulse with a higher probability of hitting one of the four optimal pulses gives rise to a less degree of speech quality degradation.

4 Experimental results

There are three experiments conducted in this work. The first is a search accuracy comparison among the full search and other search approaches. Subsequently, the second is a search load comparison among the preceding search approaches. The third is a series of objective and subjective speech quality testings among various approaches for comparison purposes. The test objects are those selected out of a speech database in Chinese

language, containing 9,650 syllables out of 100 sentences for a duration over 41 min and 495,608 subframes.

For the brevity of the following discussion, the RCM approach with N candidate pulses is abbreviated as RCM-N, $1 \leq N \leq 8$. For instance, RCM-1 symbolizes the one with merely a candidate pulse extracted out of each track. Similarly, the GPR approach with the number R of repetitions is designated as GPR-R.

Table 6 Comparison of statistics on quantized MOS-LQO scores among various methods

| MOS-LQO	G.729A	GPR-2	IFPR	Proposed method		
				$N = 2$	$N = 3$	$N = 4$
3.50	0	0	0	0	0	0
3.55	0.01	0	0	0.01	0	0
3.60	0	0.01	0	0.01	0	0
3.65	0.01	0	0.02	0.01	0.01	0
3.70	0.02	0.01	0.01	0	0.04	0.01
3.75	0.04	0.02	0.01	0.03	0.02	0.03
3.80	0.06	0.02	0.04	0.06	0	0.02
3.85	0.04	0.05	0.03	0.10	0.09	0.03
3.90	0.08	0.10	0.11	0.19	0.14	0.14
3.95	0.25	0.25	0.26	0.26	0.22	0.23
4.00	0.28	0.24	0.29	0.26	0.28	0.31
4.05	0.17	0.23	0.19	0.05	0.17	0.19
4.10	0.04	0.06	0.01	0.01	0.02	0.02
4.15	0	0	0.02	0	0	0.01
4.20	0	0.01	0	0.01	0.01	0.01
4.25	0	0	0.01	0	0	0
4.30	0	0	0	0	0	0
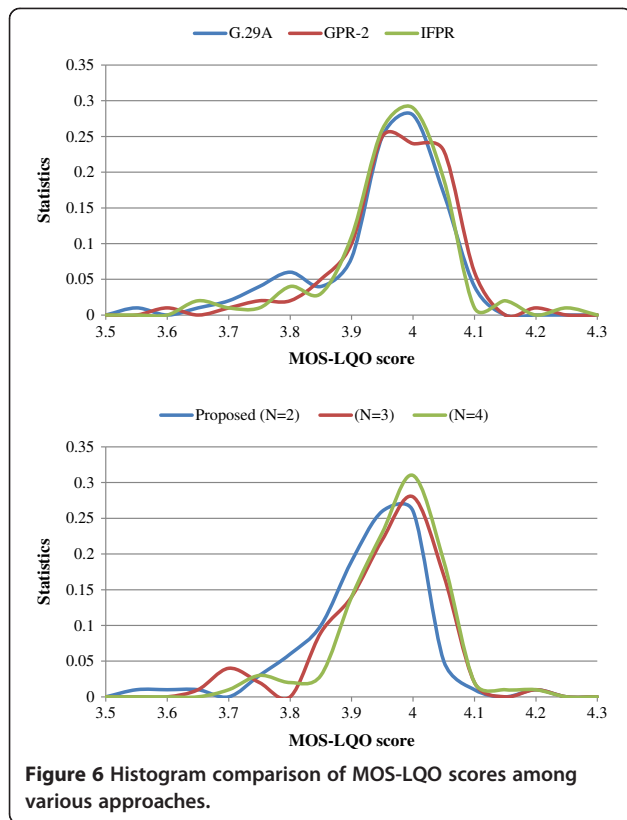
Figure 6 Histogram comparison of MOS-LQO scores among various approaches.

Tabulated in Table 3 is the search accuracy analysis among various approaches, that is, the hit probability of individual approach against the optimal pulse identified through a full search. During the search process, the best case is the one to successfully locate four intended pulses, the all right case, and the worst is to locate none, the all wrong case. As tabulated in Table 3, tracking the all right case as an instance, the accuracies made by G.729, GPR-2, IFPR, and RCM-2 are 68.6438%, 76.1053%,

68.0824%, and 50.3579%, respectively, while that by the proposed method falls between 17.3353% ($N = 1$) and 69.0717% ($N = 4$).

Tabulated in Table 4 is the comparison of the search load, that is, the number of searches performed and those required in the evaluation of Q_k defined in (2). It is found that G.729A requires 320 searches, GPR-2 64, IFPR 48, RCM-2 16, and the proposed method a number somewhere between 1 ($N = 1$) and 64 ($N = 4$). Accordingly, a search complexity is as intended reduced.

Subsequently, the values of ITU-T P.862 perceptual evaluation of speech quality (PESQ) [15] and ITU-T P.862.1 mean opinion score and listening quality objective (MOS-LQO) [16] are evaluated for objective speech quality comparison among various approaches. Table 5 gives comparisons on PESQ and MOS-LQO, each including the mean, standard deviation (STD), maximum, and minimum values, respectively. In comparison with MOS-LQO, G.729A, all the approaches provide a comparable speech quality within a 1% deviation, except that RCM-1 exhibits a 3% drop.

Moreover, there are up to 100 MOS-LQO scores measured to make a histogram. The MOS-LQO scores are firstly quantized with a maximum quantization error of 0.025. For instance, a MOS-LQO score in the range of 3.475 to 3.525 is quantized as 3.50. Table 6 gives the statistics and normalization on quantized MOS-LQO scores for comparison purposes among G.729A, GPR-2, IFPR, and this proposal. Figure 6 is a graphic presentation of Table 6 for a clear view of the MOS-LQO score distribution.

On the other hand, a forced choice listening test is conducted for a subjective speech quality comparison among the proposed ($N = 2$), the GPR-2, and the IFPR search approaches. A set of ten arbitrary sentences was selected and processed by such three approaches in each test group. Each set of processed sentences was evaluated
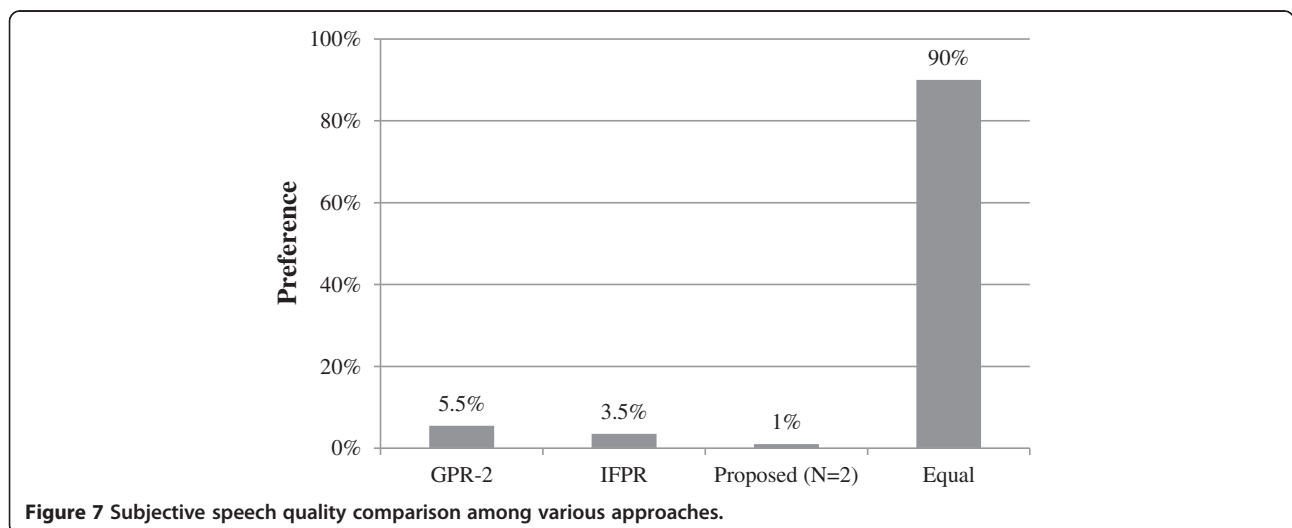

Figure 7 Subjective speech quality comparison among various approaches.

by 20 listeners. The preference score is evaluated for the three approaches by each single listener in each group. Presented in Figure 7 is a bar graph of subjective speech quality testing versus search strategy. Merely an extremely low percentage of listeners can tell a performance difference among these three search strategies, while as high as 90% of listeners cannot distinguish any differences for the test cases. This finding indicates that this proposal does provide a comparable subjective speech quality. Moreover, there is an evidence that little difference in MOS-LQO values reflects an indistinguishable subjective speech quality.

Furthermore, it is a point worthy of mention that with a marginal variation in MOS-LQO, a low level of search complexity signifies a superior system performance, e.g., the proposed method at $N = 2$ and $N = 3$. Particularly, it merely takes eight searches at $N = 2$, a figure tantamount to 12.5% of that required in GPR-2, 16.7% in IFPR, and 50% in RCM-2, but provides comparable speech quality.

5 Conclusions

An improved version of RCM approach is presented in this work as an efficient means to enhance the performance of search over an algebraic codebook when applied to a G.729A speech codec. It is experimentally demonstrated that this proposal requires as few as eight searches in the case of $N = 2$, that is, a search load tantamount to 2.5% of that in G.729A, 12.5% in GPR-2, 16.7% in IFPR, and 50% in RCM-2, but still provides a comparable speech quality. Thus, this proposal is validated as a superior candidate in the aspect of search performance. In addition, the proposed approach can be implemented to other ACELP-based speech coders.

Furthermore, this improved G.729A speech codec can be utilized to improve the VoIP performance on smart phone. As a consequence, the energy efficiency requirement is met for an extended operation time period due to computational load reduction.

Author details
[1]Department of Electrical Engineering, National Taipei University of Technology, 1, Sec. 3, Chung-hsiao E. Rd., Taipei 10608, Taiwan. [2]Department of Electrical Engineering, National Chin-Yi University of Technology, 57, Sec. 2, Zhongshan Rd., Taiping Dist, Taichung 41170, Taiwan.

References
1. JP Adoul, P Mabilleau, M Delprat, S Morissette, Fast CELP coding based on algebraic codes, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (Dallas, Texas, USA, 1987), pp. 1957–1960
2. C Laflamme, JP Adoul, R Salami, S Morissette, P Mabilleau, 16 kbps wideband speech coding technique based on algebraic CELP, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (Toronto, Canada, 1991), pp. 13–16
3. R Salami, C Laflamme, JP Adoul, A Kataoka, S Hayashi, T Moriya, C Lamblin, D Massaloux, S Proust, P Kroon, Y Shoham, Design and description of CS-ACELP: a toll quality 8 kb/s speech coder. IEEE Trans. Speech Audio Process. **6**(2), 116–130 (1998)
4. SM Tsai, JF Yang, Efficient algebraic code-excited linear predictive codebook search. IEE Proc. Vis. Image Signal Process. **153**(6), 761–768 (2006)
5. R Salami, C Laflamme, B Bessette, JP Adoul, ITU-T G.729 Annex A: reduced complexity 8 kb/s CS-ACELP codec for digital simultaneous voice and data. IEEE Commun. Mag. **35**(9), 56–63 (1997)
6. FK Chen, JF Yang, YL Yan, Candidate scheme for fast ACELP search. IEE Proc. Vis. Image Signal Process. **149**(1), 10–16 (2002)
7. H Park, Y Choi, D Lee, Efficient codebook search method for ACELP speech codecs, in *Proceedings of the IEEE Speech Coding Workshop* (Tsukuba, Ibaraki, Japan, Tsukuba, Ibaraki, Japan, 2002), pp. 17–19
8. ED Lee, MS Lee, DY Kim, Global pulse replacement method for fixed codebook search of ACELP speech codec, in *Proceedings of the 2nd IASTED International Conference on Communications, Internet and Information Technology* (Scottsdale, Arizona, USA, 2003), pp. 372–375
9. B Geiser, P Jax, P Vary, H Taddei, S Schandl, M Gartner, C Guillaume, S Ragot, Bandwidth extension for hierarchical speech and audio coding in ITU-T Rec. G.729.1. IEEE Trans. Audio Speech Lang. Process. **15**(8), 2496–2509 (2007)
10. ED Lee, SH Yun, SI Lee, JM Ahn, Iteration-free pulse replacement method for algebraic codebook search. Electron. Lett. **43**(1), 59–60 (2007)
11. ED Lee, JM Ahn, Efficient fixed codebook search method for ACELP speech codecs. Lect. Notes Comput. Sci. **4413**, 178–187 (2007)
12. CY Yeh, YJ Su, Reduced candidate mechanism for an algebraic code-excited linear-prediction codebook search. IET Commun. **6**(17), 2864–2869 (2012)
13. FK Chen, GM Chen, BK Su, YR Tsai, Unified pulse-replacement search algorithms for algebra codebooks of speech coders. IET Signal Process. **4**(6), 658–665 (2010)
14. CY Yeh, CZ Zhuo, An efficient complexity reduction algorithm for G.729 speech codec. Comput. Math. Appl. **64**(5), 887–896 (2012)
15. ITU-T Recommendation P.862, *Perceptual evaluation of speech quality (PESQ): an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*, 2001
16. ITU-T Recommendation P.862.1, *Mapping Function for Transforming P.862 raw Result Scores to MOS-LQO*, 2003