***1) Name of the workshop :*** Workshop on Multimedia Technologies

***2) Title of the paper :*** Speech Coding with Fine Granularity Scalability Based on ITU-T G.723.1

***3) A short abstract :*** A speech coding mechanism with fine granularity scalability is developed baesd on the low-rate coder of ITU-T G.723.1. The availabe bit rates range from 3.9 kbps to 5.3 kbps with a granularity of as low as 0.13 kbps. No extra overhead is added to the bitstream. And the extra efforts required in the computation is little.

***4) Authors:***

***Name :*** Fang-Chu Chen, fcchen@itri.org.tw
I-Hsien Lee

***Affiliation :*** Industrial Technology Research Institute
Computer & Communications Research Laboratories

***Address :*** Bldg. 11, 195-11 Sec. 4, Chung Hsing Rd.
Chutung, Hsinchu, Taiwan 310, R.O.C.

***Tel :*** 886-3-5914782

***Fax :*** 886-3-5829731

***Name of the contact person:*** Fang-Chu Chen , fcchen@itri.org.tw

***5) Keywords:*** speech coding, fine granularity scalability, scalable coding

# Speech Coding with Fine Granularity Scalability Based on ITU-T G.723.1

Fang-Chu Chen and I-Hsien Lee

Computer & Communications Research Laboratories
Industrial Technology Research Institute
Bldg. 11 195-11 Sec.4 Chung Hsing Rd., Chutung, Hsinchu, Taiwan 310

***Abstract*—A speech coding mechanism with fine granularity scalability is developed based on the low-rate coder of ITU-T G.723.1. The availabe bit rates range from 3.9 kbps to 5.3 kbps with a granularity of as low as 0.13 kbps. No extra overhead is added to the bitstream. And the extra efforts required in the computation is little.**

## A. Introduction

The flexibility of bandwidth usage in a transmission channel has become a major issue in this multimedia era, where the amount of data and the number of users occupying the channel are often unknown at the time of encoding. Multi-bit-rate stream source coding is one of the solutions. In accordance with this type of coding, a scalable source coder with fine granularity scalability (FGS), which requires only one set of encoding algorithm while allowing the channel and the decoder the freedom of discarding various number of bits in the bit stream, has become favored in the next generation of communication standards. A scalable bit stream consists of a base layer followed by one or more enhancement layers. The base layer is the minimum requirement and has to be received by the decoder in order to maintain an acceptable quality of the decoded contain of the stream. The enhancement layers, on the other hand, are used to improve the base-layer speech and may be ignored. In the scalable coding layered scalability requires the enhancement layers to be discarded one layer at a time, which often times is more than needed. FGS outperforms layered scalability in that the enhancement layer can be discarded with finer granularity. This feature of FGS provides the channel traffic supervisor a much easier and more flexible way to control the bandwidth used by each source stream.

General audio and video coding algorithms with FGS have been adopted as part of MPEG-4 international standard [1]. On the other hand, an FGS speech coding technique based on the popular code excited linear prediction (CELP) algorithm has not yet been standardized. The FGS algorithms used in MPEG-4 general audio and video share a common strategy, in that the enhancement layers are distinguished by the different bit significance level at which a bit plane or a bit array is sliced from the spectral residual. When a bit stream is to be shortened those bits at the end of the enhancement layer, i.e., with the least bit significance levels, will be discarded first. This method, however, may not work well for a highly parametric coder such as CELP-based ITU-T G.729, ITU-T G.723.1, GSM, and 3GPP [2][3][4][5]. The facts that all the above standard coders support multi-bit-rates, especially the Adaptive Multi-Rate (AMR) supported by GSM and 3GPP, indicates that speech coding needs a mechanism for easy bit rate adaptation as well. The advantages of better and more flexible bit rate adaptation offered by FGS coding can be proved useful. It is therefore the purpose of this article to develop a CELP based FGS speech coding process in order to extend the scope of FGS to speech applications. For easy referencing and straight demonstrating the low-rate coder of ITU-T G.723.1 will be used as the basis of such development.

## B. Method

1) Basics of CELP

In a CELP-based speech coder, a human vocal track is modeled as an all-pole filter by the technique of linear predictive coding (LPC) and is responsible for vowels. On the other hand, a glottal vibration is modeled as an periodic excitation vector for this LPC filer and is responsible for pitch. Under this LPC model it is expected that if the pitch excitation vector and the LPC filter are well coded the signal obtained by filtering the pitch excitation vector through the LPC filter can symthesized any speech one demands. However, this simple model always leaves errors between the synthesized speech and the original one. In the standards, the errors due to the imperfections of the model or the LPC/pitch coding are, to a great extent, compensated for with stochastic process. The stochastic process is often time implemented by fixed-code pulses which are added to the pitch part of the excitation in order that when the combined excitation vector is filtered through the LPC filter the errors can be minimized. Alternatively speaking, speech component generated by the fixed-code pulses is used to enhance the quality of that by the simple LPC speech synthisis model.

For each speech signal to be encoded, the stream is partitioned into frames and further into some even number of subframes. During the encoding process the parameters associated with LPC filtering and the fixed-code pulses are searched through an analysis-by-synthesis method on a frame/subframe basis. These parameters are then sent to the decoder in order for obtaining a synthesized speech best resembling the original one. According to CHEN[6], the number of the fixed-code pulses, which occupies a big percentage of the total bit rate, can be cut in half by removing those pulses in the odd-numbered subframes. Using the low-rate coder of ITU-T G.723.1 as an example, the method leads to a 27% reduction in the bit rate with only 1 dB SEGSNR (segmental signal-to-noise ratio) deterioration in the decoded speech. Based on this previous study, FGS of ITU-T G.723.1 can be achieved by delicately adding back the pulses of the odd-numbered subframes, in other words, by placing the information bits associated with the fixed-code pulses of the odd-numbered subframes in the enhancement layer. The following sections described the details of the modifications involved in realizing this concept.

2) Modifications on the algorithm

The enhancement layer of an FGS bit stream is allowed to be discarded as a whole or by part depending on the transmission environment. Placing the odd-numbered subframe pulses in the enhancement layer implies that the number of those pulses received by the decoder is unknown at the encoder side. This jeopardizes the analysis-by-synthesis method used in the standard coder for the following reasons: The purpose of the analysis-by-synthesis method, by imbedding a decoder in the encoding process, is for the encoder to foresee the exact speech decoded by the decoder on the other end of the transmission line. If the encoder has no knowledge about the number of odd-numbered subframe pulses actually used by the decoder it would have no base for constructing the best parameters to be sent to the decoder. This is inevitable for a scalable coding. One way to minimize this problem is to assume the worst case of the receiving condition, i.e., always assume that the decoder receives none of the information bits from the enhancement layer. To be more precise in terms of implementation, the excitation vector and the memory states (of the LPC filtering) passed over from an odd-numbered subframe to the next even-numbered subframe have to be constructed without any information from the odd-numbered subframe pulses (Fig. 1). The odd-numbered subframe pulses are still searched and generated, they, however, are purely used for extra quality enhancement of that subframe

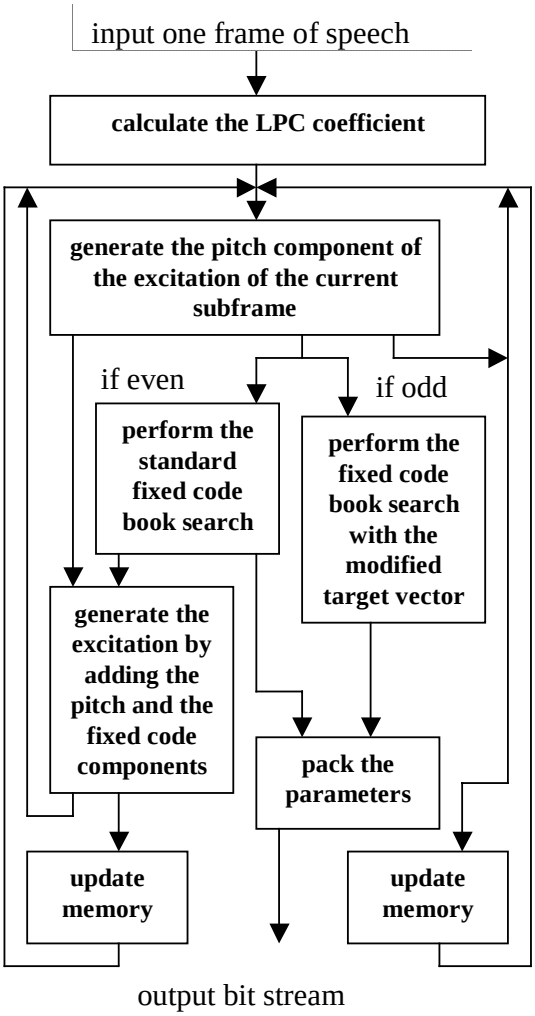


Fig.1 The flow chart of the modified encoder

and are never recycled in the future subframes. If the encoder is allowed to recycle any of the odd-numbered subframe pulses which are not received by the decoder then the codes selected for the next subframe might not be the right choice for the decoder and an error would occur. The same rule applies to the decoder (Fig. 2). That is, when updating the excitation vectors or the memory states the components generated by any odd-numbered subframe pulses have to be completely removed.

The worst-case-assumption described above ensures the performance of the analysis-by-synthesis method used in the encoder, it, however, inevitably introduces certain degree of subframe boundary discontinuity at the decoder side. The problem is due to the fact that the odd-numbered subframe pulses are not used for memory updating, meaning that the speech components generated by those pulses for extra enhancements are not fed back to the LPC synthesizer at the odd-numbered/even-numbered subframe boundaries.

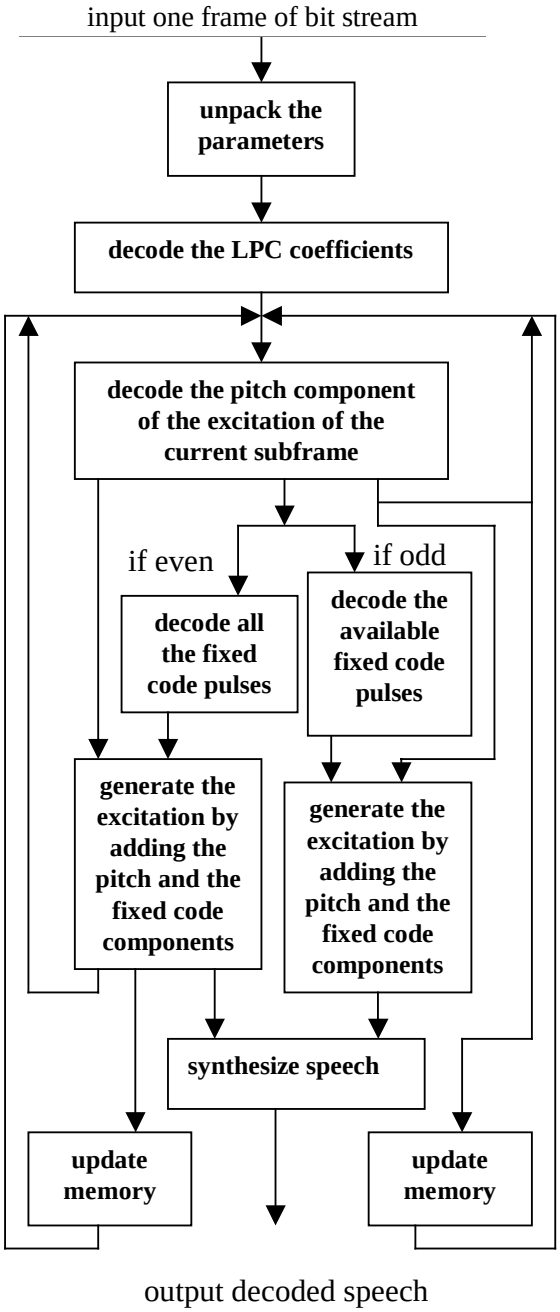


Fig.2 The flow chart of the modified decoder

Obviously, one has to minimize this effects of the speech components generated by the non-recycled pulses on the following even-numbered subframe. Fortunately, since only ten speech samples from the previous subframe are needed in a tenth-order LPC synthesizer only the last ten samples of the subframe needs to be considered. As a matter of fact it calls for only a minor change in the algorithm to serve this purpose as will be described right below. Since the LPC-filtered pulses are chosen to best mimic a target signal, one way to reduce the effects of the pulses is, therefore, to linearly taper off the magnitude of the last ten samples of the target vector prior to the fixed code search for each odd-numbered subframe. This simple modification on the target vector not only reduces the effects on the last ten samples of an odd-numbered subframe it, at the same time, prevents a daunting attempt of breaking up the integrity of the well-established fixed codebook search algorithm.

3) Modifications on the bit ordering

A full-length bit stream contains a base layer and a full-length enhancement layer. A full-length enhancement layer, according to the previous section, contains all the pulses in the odd-numbered subframes of a frame. Since the basic structure of the coder remains the same even after the modifications, the number of total bits in a full-length bit stream of a frame is the same as that of a standard coder. The bit order, however, has to be modified in order to accommodate the ability of flexible bit rate transmission. The criterion is to transmit those bits needed in the base layer before those for the enhancement layer. Moreover, the bits for the pulses of one odd-numbered subframe are grouped together. Since all four pulses of each subframe share the same grid and gain the grid and gain bits will be placed before those of positions and signs. The bits for positions and signs will be broken up and reassembled so that three position bits and one sign bit of each pulse will be put together. With this ordering pulses are abandoned in the way that those in the same odd-numbered subframe are discarded first before those in the other odd-numbered subframe are affected. Table 1 shows one example of the bit reordering of the low bit rate coder of ITU-T G.723.1. Note that, except for the bit order the bit fields and the total bit number remain the same as those defined in G.723.1. In other words, no extra overhead is introduced. In this table only those bits in the dark shaded area are assembled in the same manner as that used in the standard. The new order of the rest of the bits is such that those bits in the light shaded area, together with the bits in the dark shaded area, constitute the base layer, and those 42 bits in the unshaded area constitute the enhancement layer. With the modified encoding algorithm the encoder encodes and provides the full-length bitstream to a channel supervisor. This supervisor can discard up to 42 bits from the end of the bitstream depending on the channel traffic. Then, according to the number of the bits received, the decoder at the other end of the channel decodes the bitstream on the a pulse's basis, meaning that if the number of the enhancement bits received is not enough to decode one specific pulse then that pulse will be abandoned.

TABLE 1.
Bit reordering table from the low-rate coder of ITU-T G.723.1

| Transmitted octets | Bit order |
|---|---|
| 1 | LPC_B5…LPC_B0,VADFLAG_B0,RATEFLAG_B0 |
| 2 | LPC_B13…,LPC_B6 |
| 3 | LPC_B21…LPC_B14 |
| 4 | ACL0_B5…ACL0_B0,LPC_B23,LPC_B22 |
| 5 | ACL2_B4…ACL2_B0,ACL1_B1,ACL1_B0,ACL0_B6 |
| 6 | GAIN0_B3…GAIN0_B0, ACL3_B1,ACL3_B0,ACL2_B6,ACL2_B5 |
| 7 | GAIN0_B11…GAIN0_B4 |
| 8 | GAIN1_B11…GAIN1_B4 |
| 9 | GAIN2_B7…GAIN2_B0 |
| 10 | GAIN3_B7…GAIN3_B4,GAIN2_B11…GAIN2_B8 |
| 11 | PSIG0_B1, PSIG0_B0, GRID2_B0, GRID0_B0, GAIN3_B11…GAIN3_B8 |
| 12 | POS0_B1, POS0_B0, PSIG2_B3…PSIG2_B0, PSIG0_B3, PSIG0_B2 |
| 13 | POS0_B9…POS0_B2, |
| 14 | POS2_B5…POS2_B0, POS0_B11, POS0_B10 |
| 15-1 | POS2_B11…POS2_B6 |
| 15-2 | **GAIN1_B1, GAIN1_B0** |
| 16 | **PSIG1_B1,POS1_B2…POS1_B0, PSIG1_B0, GRID1_B0, GAIN1_B3, GAIN1_B2** |
| 17 | **PSIG1_B3, POS1_B8…POS1_B6, PSIG1_B2, POS1_B5…POS1_B3,** |
| 18 | **GRID3_B0,GAIN3_B3…GAIN3_B0, POS1_B11…POS1_B9** |
| 19 | **POS3_B5…POS3_B3, PSIG3_B1, POS3_B2…POS3_B0, PSIG3_B0,** |
| 20 | **POS3_B11…POS3_B9, PSIG3_B3, POS3_B8…POS3_B6, PSIG3_B2,** |

Roughly speaking, this leads to a granularity of 4 bits for the position and sign bits of the last three pulses in each odd-numbered subframe. The granularity is 9 bit if the first pulse is to be discarded as well, since, when all four pulses are abandoned, there will be no need to retain the gain and the grid bits. This is equivalent to a granularity of about 0.13 kbps/0.3 kbps within the bit rate range from 3.9 kbps to 5.3kbps.

### C. Results

An FSG speech coding technique designed from the method described in the previous sections involves only very little modifications from the standard methods. It takes almost the same computation load and generates exactly the same length of a bit stream. Theoretically, the worst case of the speech quality decoded by such a FGS scalable coder is that with all 42 enhancement bits being discarded. As pulses are added back the speech quality is expected to improve. To demonstrate this claim, a performance curve is shown in Fig. 3 where the SEGSNR values of each decoded speech with reference to that using the standard low-rate coder of the ITU-T G.723.1 are plotted. The test input is the same 53-second speech used in the reference of [6]. The abscissa of Fig. 3 represents the total number of pulses used in subframes 1 and 3 (the same for all frames.) With each odd-numbered subframe being allowed four pulses in the standard and the manner the bits are assembled in table 1, if the total number of pulses is shy of eight but greater than four then the missing pulses are from subframe 3. If the total number of odd-numbered subframe pulses are less than four then they are all from subframe 1. In the worst case when the pulse number is zero it indicates that no pulses are used in any odd-numbered subframe. This graph clearly demonstrates that the speech quality depends on the number of enhancement bits available in the decoder. This is exactly the behavior one expects from a scalable speech coder.

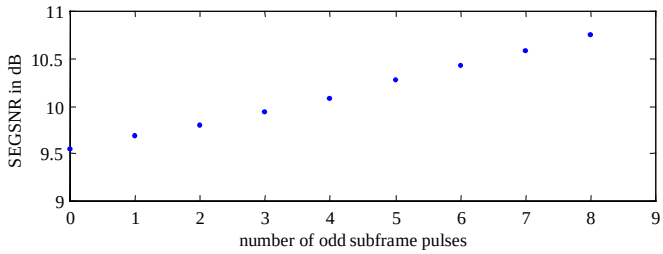


Fig. 3 Performances among difference numbers of odd-numbered subframes pulses used in the decoder.

### References


[1] ISO/IEC 14496, the MPEG-4 standard.

[2] ITU-T Recommendation G.723.1: 'Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s.'

[3] ITU-T Recommendation G.729: 'Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACLEP),' Annex D '6.4 kbit/s CS-ACELP speech coding algorithm,' Annex E '11.8 kbit/s CS-ACELP speech coding algorithm.'

[4] ETS EN 301 704, 'Digital cellular telecommunications system; Adaptive Multi-Rate (AMR) speech transcoding.'

[5] 3GPP TS 26.190, ‘Speech Codec speech processing functions; AMR Wideband speech codec; Transcoding functions (Release 5).’

[6] Chen, Fang-Chu ,“Suggested new bit rates for ITU-T G.723.1”, Electronics Letters Vol. 35 No.18 p. 1523, 1999.