G723.1 Standard

This coder is based on the same principles as the CELP coder, that is, it is based on the principles of linear predictive analysis-by-synthesis coding and attempts to minimize a perceptually weighted error signal. However it has some distinguishing features.

The G.723.1 coder operates on frames of 240 samples each (30 ms at an 8 kHz sampling rate). Each block is divided into 4 subframes of 60 samples each.

• For every subframe a 10th order linear prediction filter is computed using the original speech signal. The LPC are transformed to the LSP. The set of LSP for the last subframe is quantized using vector quantizer. Due to using vector quantization only 24 bits are required for LSP instead of 34 bits in the CELP coder.
LSP quantizer

\[
p_n^T = [p_{1,n} p_{2,n} \ldots p_{10,n}] \quad \text{DC removed vector of LSPs}
\]

\[
\bar{p}_n = b[\tilde{p}_{n-1} - p_{\text{DC}}] \quad \text{Prediction from the previously decoded LSP vector, } b=12/32
\]

\[
e_n = p_n - \bar{p}_n \quad \text{the residual LSP error.}
\]

3 codebooks (3,3,4 components of \(e_n\)) of size 256 codewords each are used for quantization. We minimize the weighted error

\[
E_{l,m} = (p'_m - \tilde{p}_{l,m})^T W_m (p'_m - \tilde{p}_{l,m}), \quad 0 \leq m \leq 2, \quad 1 \leq l \leq 256
\]

\[
\tilde{p}_{l,m} = \bar{p}_m + p_{\text{DC}_m} + e_{l,m}, \quad 0 \leq m \leq 2, \quad 1 \leq l \leq 256
\]

\[
p' = p + p_{\text{DC}}
\]
LSP dequantizer

The three sub-vectors \( \{e_{l_m,m,n}\} \) for \( m=0..2 \) form \( \tilde{e}_n \)

\[
\tilde{p}_n = \overline{p}_n + \tilde{e}_n + p_{DC}
\]

A stability check is performed on the decoded LSP vector

\[
\tilde{p}_{i+1,n} - \tilde{p}_{i,n} \geq \Delta_{\text{min}} = 62.5 \text{ Hz}, 1 \leq i \leq 9
\]

Otherwise vector components are modified as

\[
\tilde{p}_i = \tilde{p}_i - \Delta_{\text{min}} / 2 \quad \tilde{p}_{i+1} = \tilde{p}_{i+1} + \Delta_{\text{min}} / 2
\]

If after 10 iterations the condition of stability is not met, the previous LSP vector is used.
LSP interpolation

\[ \tilde{p}_{ni} = \begin{cases} 
0.75\tilde{p}_{n-1} + 0.25\tilde{p}_n, & i = 0 \\
0.5\tilde{p}_{n-1} + 0.5\tilde{p}_n, & i = 1 \\
0.25\tilde{p}_{n-1} + 0.75\tilde{p}_n, & i = 2 \\
\tilde{p}_n, & i = 3
\end{cases} \]
Both the quantized and the unquantized LPC are used to construct the short-term perceptual weighting filter, which is used to filter the entire frame and to obtain the perceptually weighted speech signal.

• To speed up a search over the adaptive codebook the following approach is used. For every two sub-frames (120 samples), the open loop pitch period is computed:

\[
\max_j \frac{\left( \sum_{i=0}^{119} s(n+i) \cdot s(n+i-j) \right)^2}{\sum_{i=0}^{119} s(n+i-j)^2}, 18 \leq j \leq 145
\]
G723.1

Using the pitch predictor that is a linear FIR filter of 5\textsuperscript{th} order given by its pulse response \((\alpha_{-2}, \alpha_{-1}, \alpha_0, \alpha_1, \alpha_2)\) the optimal prediction of the periodical part of the speech signal is computed by minimizing

\[
\min_{j,a} = \sum_{n=0}^{59} \left( s(n) - \sum_{i=-2}^{i=2} \alpha_i s(n - p + i - j) \right)^2 ,
\]

where \( p = j_{opt} \) is the pitch estimate.

For subframes 0, 2 \( j = \pm 1 \) and \( p + j \) is transmitted using 7 bits. For subframes 1,3 \( p + j \) is coded differentially using 2 bits and may differ from the previous subframe pitch lag only by \(-1, 0, +1, \) or \(+2\).

The coefficients \((\alpha_{-2}, \alpha_{-1}, \alpha_0, \alpha_1, \alpha_2)\) are vector quantized using two codebooks with 85 or 170 entries for high rate and 170 entries for low rate.
The contribution of the pitch predictor $p(n), n = 0,\ldots,59$ is subtracted from the target vector $s(n), n = 0,\ldots,59$ to obtain the residual signal: $r(n) = s(n) - p(n), n = 0,\ldots,59$

The non-periodic component of the excitation is approximated using the obtained residual signal. For the high bit rate, multipulse maximum likelihood quantization (MP-MLQ) excitation is used, and for the low bit rate, an algebraic codebook excitation (ACELP) is used.

The residual signal $r(n), n = 0,\ldots,59$ is approximated as follows:

$$r'(n) = \sum_{j=0}^{n} h(j) \cdot v(n - j), 0 \leq n \leq 59,$$

where $v(n)$ is the excitation of the filter with pulse response $h(n): v(n) = G \sum_{m=0}^{M-1} \beta_m \delta(n - l_m), 0 \leq n \leq 59$
G723.1

$G$ is the gain factor, $\delta(n)$ is a Kronecker delta-function, $\beta_m$ and $l_m$ are the signs ($\pm 1$) and the positions of Kronecker functions, respectively, and $M$ is the number of pulses.

$M = 6$ for even subframes and it is 5 for odd subframes.

The positions can be either all odd or all even. This is indicated by 1 bit.

The problem is to estimate the unknown parameters $G$, $\beta_m$, $l_m$, $m = 0, \ldots, M - 1$ that minimize the squared error

$$\sum_{n=0}^{59} (err(n))^2 = \sum_{n=0}^{59} (r(n) - r'(n))^2 = \sum_{n=0}^{59} \left( r(n) - G \sum_{m=0}^{M-1} \beta_m h(n - l_m) \right)^2$$
G723.1

The following gain estimate is computed:

\[
G_{\text{max}} = \max_{j=0,1,\ldots,59} \left\{ \left( \sum_{n=j}^{59} r(n)h(n-j) \right) \right\} \left( \sum_{n=0}^{59} (h(n))^2 \right) .
\]

The estimated gain \( G_{\text{max}} \) is scalar quantized with cell size 3.2 dB (nonuniform quantizer with 24 cells). Around the approximating value, \( \tilde{G}_{\text{max}} \), additional gain values are selected within the range \( [\tilde{G}_{\text{max}} - 3.2, \tilde{G}_{\text{max}} + 6.4] \text{dB} \).

For each of these gain values the signs and locations of the pulses are estimated and quantized. Finally, the combination of the quantized parameters that yields the minimum of \( \sum_{n=0}^{59} (\text{err}(n))^2 \) is selected. The best gain approximation and the optimal pulse locations and signs are transmitted.
Combinatorial coding

Combinatorial coding is used to transmit the pulse locations.

Assume that we consider binary sequences of length $L$ and weight $w$ it is easy to compute that there are

$$N = \binom{L}{w}$$
such sequences. The combinatorial coding spend

$$\log_2 N$$
bits to transmit the number of sequence.

For example, let $L = 30$ and $w = 6$ then

$$N = \binom{30}{6} = 593775$$

and

$$\lceil \log_2 N \rceil = 20$$
bits.
The standard provides two bit rates 5.3 kb/s and 6.4 kb/s.

For rate 5.3 kb/s a 17-bit algebraic codebook is used to innovate excitation \( \nu(n) \). The innovation vector contains at most 4 non-zero pulses. The amplitudes and positions are chosen from the admissible values given in the Table. The positions of all pulses can be simultaneously shifted by one (to occupy odd positions) which needs 1 extra bit.

Each pulse position is encoded with 3 bits and each pulse sign is encoded in 1 bit.

This gives a total of 16 bits for the pulses. An extra bit is used to encode the shift resulting in a 17-bit codebook.
### ACELP pulse positions

<table>
<thead>
<tr>
<th>Amplitude</th>
<th>Positions</th>
</tr>
</thead>
<tbody>
<tr>
<td>± 1</td>
<td>0, 8, 16, 24, 32, 40, 48, 56</td>
</tr>
<tr>
<td>± 1</td>
<td>2, 10, 18, 26, 34, 42, 50, 58</td>
</tr>
<tr>
<td>± 1</td>
<td>4, 12, 20, 28, 36, 44, 52, (60)</td>
</tr>
<tr>
<td>± 1</td>
<td>6, 14, 22, 30, 38, 46, 54, (62)</td>
</tr>
</tbody>
</table>
ACELP

The codebook is searched by minimizing the MSE between the residual signal \( r(n) \) obtained after subtracting the pitch contribution from the original speech, and the filtered weighted codeword \( v(n) \) from the algebraic codebook

\[
\sum_{n=0}^{59} \left( r(n) - G \sum_{j=0}^{n} h(j)v(n - j) \right)^2
\]

where \( h(n) \) is the pulse response of the filter, \( G \) is the codebook gain.
## Bit allocation for the 6.4 kb/s coding algorithm

<table>
<thead>
<tr>
<th>Parameters coded</th>
<th>Sub-frame 0</th>
<th>Sub-frame 1</th>
<th>Sub-frame 2</th>
<th>Sub-frame 3</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPC indices</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>Adaptive codebook lags</td>
<td>7</td>
<td>2</td>
<td>7</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td>All the gains combined</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>48</td>
</tr>
<tr>
<td>Pulse positions</td>
<td>20</td>
<td>18</td>
<td>20</td>
<td>18</td>
<td>76</td>
</tr>
<tr>
<td>Pulse signs</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>22</td>
</tr>
<tr>
<td>Grid index</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td><strong>TOTAL:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>192</td>
</tr>
</tbody>
</table>
## Bit allocation of the 5.27 kb/s coding algorithm

<table>
<thead>
<tr>
<th>Parameters coded</th>
<th>Sub-frame 0</th>
<th>Sub-frame 1</th>
<th>Sub-frame 2</th>
<th>Sub-frame 3</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPC indices</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>Adaptive codebook lags</td>
<td>7</td>
<td>2</td>
<td>7</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td>All the gains combined</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>48</td>
</tr>
<tr>
<td>Pulse positions</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>48</td>
</tr>
<tr>
<td>Pulse signs</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>Grid index</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>TOTAL:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>158</td>
</tr>
</tbody>
</table>
G.723.1

The major differences between two rates are in the pulse positions and amplitudes coding. Also at the lower rate only 170 codebook entries are used for the gain vector of the long term predictor.

Bit rate for highrate coder is: \((8000/240)192=6.4\text{ kb/s}\)

Bit rate for lowrate coder is: \((8000/240)158=5.27\text{ kb/s}\).
Other CELP-like coders

G.728 is a low-delay CELP. It uses a **backward adaptive** LP filter that is updated every 2.5 ms. The filter coefficients are not transmitted. At the decoder side the filter is constructed using synthesized speech signal.

The 10\textsuperscript{th} order gain predictor is used and there are 1024 excitation vectors which are split into 4 possible gains (quantized differences between gains and their predictions), 2 possible signs and 128 possible shapes. Excitation is transmitted for each frame containing 5 samples.

Bit rate is equal to \( \frac{8000}{5 \times (2 + 1 + 7)} = 16 \) kb/s.

**IS-54 (VSELP), PDC(VSELP) coders.** They represent **vector sum excited** linear prediction coders. Non-periodical excitation is linear combination of vectors from 2 highly structured codebooks.
Other CELP-like coders

GSM standard uses RPE-LTP (regular-pulse excitation- long-term prediction) coder. Operates at 13 kb/s.

Uses frames of 160 samples and subframes of 40 samples. LP filter of order 8 is constructed. Its LSPs are encoded by 36 bits.

Periodical part is represented by gain and pitch and requires (7+2)4=36 bits for representation.

Non-periodical part of excitation is approximated as a sequence of multiple uniformly, i.e., (pitch-spaced) pulses. Residual is decimated into 3 subsequences of length 13 samples. Phase of the chosen group (with highest energy) is represented by 2 bits and maximum amplitude by 6 bits. Each pulse of this sequence has an amplitude quantized by 3 bits.

\[(13 \times 3 + 2 + 6) \times 4 = 188\] for excitation +36 for LSPs+36 for pitch part=260 bits, i.e., 260/20ms=13kb/s
MELP standard

Typically the LPC vocoders use two voicing states: voiced and unvoiced (pulse or noise excitation)
Main features of MELP vocoder are
1. Converting LPC to LSP
2. Splitting the voice band into 5 subbands for noise/pulse mixed excitation
3. Shaping pulse excitation
4. Using aperiodic pulses for transition regions between voiced and unvoiced speech segments

Provides bitrate 2.4 kb/s
Image coding standards

We use the term image for still pictures. Image coding includes compressing so-called bi-level or fax images, photographs (continuous-tone colour or monochrome images), and document images.

To compress image it is necessary to remove any redundancies observable in the signal. The obvious form of signal redundancy in most images is spatial redundancy.

Spatial redundancy takes a variety of different forms in an image, including correlations in the background image (e.g., a repeated pattern in a background wallpaper of a scene), correlations across an image (repeated occurrences of base shapes, colours, patterns, and so on).
Spatial redundancy

Repeated pattern on the background
Image coding

A variety of techniques used in modern image compression standards to compensate spatial redundancy is based on transform coding.

The second basic principle of image coding is to exploit the human’s ability essentially to pay no attention to various types of image distortions. By understanding the masking properties of the human visual system it is possible to make distortions perceptually invisible.
Coding of bi-level fax images: JBIG Standards
Each pixel of fax images can be white or black. These two pixel values can be represented by only one bit: 0 corresponds to black pixel and 1 corresponds to white pixel.

Group of successive pixels of the same colour we call run.

The binary representation of the given line is 100000010100.

G3 fax standard uses a 1-D run-length coding of pixels on each line followed by Huffman coding.

For our example we obtain (1,1) (6,0) (1,1) (1,0) (1,1) (2,0), where the first element of the pair is a run length and the second is the pixel value. This sequence of pairs (run length, value) is coded by the 2-D Huffman code.
The **G4 fax standard** provides (on average) an improvement over G3 by using **2-D run-length coding** to take advantage of vertical spatial redundancy, as well as the horizontal spatial redundancy used in G3 fax coding.

G4 uses the pixels of the **previous line as predicted values** for the pixels of the **current line**. It is performed as follows. Starting with the second line the sum modulo 2 of previous and current lines is computed. The obtained sum is processed by run-length coder followed by Huffman coding.

**G3,G4** provide good compression for simple text-based faxes and they give small compression ratios for handwritten text or binary halftone images (gray-scale images converted into binary form).
Coding bi-level images

**Binary halftone (gray-scale) images** have statistical properties that are very different from binary text and therefore need a significantly different coding algorithm to provide high-quality encoding at significant compression ratios.

**Halftone (gray-scale) images converted into dots** (as in newspapers) contain rather short runs and need significantly larger region of support for prediction than that needed for text images. **Halftone images represented by bit-planes** contain rather short runs as well.

**JBIG-1** standard provides compression ratios that are comparable to G4 fax for text sequences and it significantly improves compression for binary halftone images.
JBIG-1 standard

• JBIG-1 has a **progressive mode** that provides a low-resolution base-image and a sequence of delta files each of which corresponds to another level of resolution (each delta file doubles both the vertical and the horizontal resolution).

![Diagram of JBIG encoder](image-url)
JBIG-1 standard

Resolution reduction

Typical prediction (Diff.)

Determin. prediction

Adaptive Templates

Model Template

Adaptive arithmetic encoder

$I_D$  \rightarrow  $I_{D-1}$

$C_D$

JBIG differential level
JBIG-1 standard

• The resolution reduction block decimates $I_D$ to obtain $I_{D-1}$

• Remaining blocks implement the compression algorithm for $I_D$

The most important parts are an adaptive arithmetic coder and the model-templates block. They provide context arithmetic coding for pixels of $I_D$. For differential-layer coding context (an integer $S$) is determined by 6 pixels in the causal high-resolution image, 4 already available pixels in low-resolution image and also 4 possible spatial phase of the pixel to be encoded.

There are $4 \times 2^6 \times 2^4 = 4096$ possible contexts.

For bottom-layer encoder we have 1024 possible contexts which are formed by 9 adjacent already decoded pixels and 1 spatially separated from others.
Resolution reduction
Context arithmetic coding
Context arithmetic coding
JBIG-1 standard

- **Adaptive templates block** finds periodicities in the image and chooses symbol “A” taking into account these periodicities.

- **Differential-layer typical prediction block** looks for regions of solid color and transmits it by a special flag. None of the processing normally is done in the deterministic prediction, adaptive templates, model templates or arithmetic coding blocks.

- **Bottom-layer typical predictor** also tries to exploit solid regions of image to save processing efforts. But it is line-skipping algorithm. A line is typical if it is identical to the line above. Flag “typical” is sent. The encoder and the decoder skip the coding of all pixels in typical lines and generate them by repetition.
JBIG-1 standard

$I_0$

Typical prediction (Bottom) → Adaptive Templates → Model Template → Adaptive arithmetic encoder

JBIG bottom-layer encoder