INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.263

Appendix III

(06/2001)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Coding of moving video

---

Video coding for low bit rate communication

Appendix III: Examples for H.263 encoder/decoder implementations

ITU-T Recommendation H.263 – Appendix III

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	H.230–H.239
Communication procedures	H.240–H.259
Coding of moving video	**H.260–H.279**
Related systems aspects	H.280–H.299
SYSTEMS AND TERMINAL EQUIPMENT FOR AUDIOVISUAL SERVICES	H.300–H.399
SUPPLEMENTARY SERVICES FOR MULTIMEDIA	H.450–H.499

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation H.263

Video coding for low bit rate communication

APPENDIX III

Examples for H.263 encoder/decoder implementations

Summary

This informative Appendix III to ITU-T H.263 contains several examples of encoder and decoder implementations for information to the users of ITU-T H.263.

Source

Appendix III to ITU-T Recommendation H.263 was prepared by ITU-T Study Group 16 (2001-2004) and approved under the WTSA Resolution 1 procedure on 8 June 2001.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

ITU-T Recommendation H.263

Video coding for low bit rate communication

APPENDIX III

Examples for H.263 encoder/decoder implementations

III.1 Introduction

This appendix describes the Test Model Near-Term (TMN) for ITU-T H.263 Version 3 (including Annexes A through W) and also the subsequently approved Annex X. The purpose of the Test Model is to help manufacturers understand how to use the syntax and the decoder specification of the Recommendation via implementation examples of a video encoder.

H.263 defines bitstream syntax and a corresponding video decoder such that video terminals from different manufacturers can interoperate. The design of the video encoder is beyond the normative scope of H.263 and left to the manufacturer. However, in developing the Recommendation, preferred encoding methods have emerged. These methods produce good results in terms of video quality and compression efficiency at complexity levels suitable for operation on general and special purpose processors and for different transmissions scenarios (e.g. circuit-switched or packet-switched networks). Moreover, the performance levels obtained by these methods often serve as a benchmark for research and development of future ITU-T video coding Recommendations. This appendix describes encoder operations as well as decoder operations beyond the normative text of H.263 applicable for example in case of syntax violations due to transmission over error-prone channels.

All documents referenced in this appendix fall into one of the following categories:

- ITU Recommendations: These are referenced by their well-known shortcuts, e.g. H.323 or BT.601. In certain cases the version of the Recommendation is significant and will be included in an appropriate form, such as publication year or version number. ITU-T Recommendations can be obtained directly from ITU-T. See http://www.itu.int for details.
- ITU-T SG 16 video coding experts group contributions: These are referenced by their shortcut name, e.g. [Q15-D-58]. For most of the contributions referenced in this appendix, hyperlinks are used to point to the documents on a Q.6/16 informal ftp site currently located at ftp://standard.pictel.com/video-site.
- Other academic publications: These are referenced using an abbreviation string based primarily on authors' names and date of publication, and can be found in the corresponding cited publication.

The design of H.263 is based on a block-based motion-compensated hybrid transform video coder, consisting of motion estimation and compensation, DCT, quantization, run-length coding, VLC, as well as FLC coding. Additionally, there are several optional modes of operation permitted by H.263, as defined by Annexes in this Recommendation. This appendix assumes some familiarity with H.263, its optional modes, and video coding in general. A tutorial on H.263 Version 1 and its optional modes can be found in [GFS97], while a corresponding publication for H.263 Version 2 has been provided in [Q15-D-58].

This appendix describes encoding methods through functional units typically employed by an encoder. In addition, this appendix provides information on the application of the various methods for specific scenarios, referred to as application scenarios. These are:

- variable bit-rate coding with fixed quantizer value (see 4.2.4/H.263) for transmission over error-free channels (commonly used for video coding research purposes);

- fixed bit-rate coding in a practically error-free environment (H.320/H.324);
- fixed bit-rate coding in a packet-switched lossy environment (H.323); and
- fixed bit-rate coding in a highly bit error-prone environment (Annex C/H.324).

The application scenarios are discussed in III.2, which makes reference to mechanisms described in later clauses. Clause III.2 also defines the simulation environments assumed in this appendix. Mechanisms common to all application scenarios are described in III.3. These include motion vector (MV) search, quantization, and the use of optional modes of H.263 that are common to all application scenarios. Mechanisms relevant to specific application scenarios are discussed in III.4. Combined with the Recommendation itself, these clauses define a framework for an H.263 codec that performs reasonably well for the various application scenarios. Clause III.5 elaborates on decoder post-processing. Clause III.6 captures all information that was deemed valuable and adopted for inclusion into this appendix, but does not yet fit within any of the application scenarios.

III.2 Application scenarios

This clause discusses the considered application scenarios for H.263-based video coding. Its purpose is to provide an overview of useful mechanisms for each scenario and to outline the corresponding simulation environment where appropriate. Description of the environment outside the video data path is kept to a minimum, although references are provided for convenience.

III.2.1 Variable bit-rate coding for transmission error-free channels

The variable bit-rate coding scenario employs a constant quantizer value for all pictures and picture regions to produce a constant level of quality. This scenario is useful for video coding research and related standardization work. For example, it provides a framework for assessing the objective and subjective quality of bitstreams generated by new proposals on coding efficiency. There are two different models for this scenario: the low complexity model and the high quality model. These models are described in the next two subclauses.

III.2.1.1 Low complexity coding model

The low complexity coding model employs the optional modes defined in Profile 1 of Annex X of H.263, which are the advanced INTRA coding mode (Annex I), the deblocking filter mode (Annex J), the modified quantization mode (Annex T), and a part of the supplementary enhancement information mode (Annex L). Note that none of the enhanced capabilities of the supplementary enhancement information mode are currently discussed in this appendix. However, it should be noted that they address features that might be important for certain product designs. Motion estimation and mode decision are performed using the low complexity motion vector search and mode decision methods. Details can be found in clauses III.3.1.2 (low complexity motion vector search), III.4.1.2 (low complexity mode decision), III.3.4 (advanced INTRA coding mode), and III.3.5 (modified quantization mode).

III.2.1.2 High quality coding model

The high quality coding model is designed to provide improved reconstructed picture quality when compared to the low complexity model at the expense of increased computational complexity. The high quality coding model employs the optional modes defined in Profile 5 of Annex X of H.263, which include the modes of Profile 1 that is employed for the low complexity coding model as well as the Unrestricted Motion Vector mode (Annex D), the Advanced Prediction mode (Annex F), and the Enhanced Reference Picture Selection mode (Annex U) in case more than one reference picture is used for motion compensation. Motion estimation and mode decision are performed using the high quality coding motion vector search and mode decision methods. Details can be found in clauses III.3.1.4 (high quality coding motion vector search), III.4.1.3 (high quality coding mode decision), III.3.4 (advanced INTRA coding mode), III.3.5 (modified quantization mode), and

III.4.2.4 (combined mode decision, motion vector search, and Enhanced Reference Picture Selection).

III.2.2 Fixed bit-rate coding for transmission over error-free channels (H.320 or H.324)

This application is characterized by the need to achieve a fixed target bit rate with reasonably low delay. The transport mechanism is bit-oriented, and provides an environment that can be considered error-free in all practical cases. This scenario is, compared to the previous one, closer to a practical application, thus the complexity versus quality trade-off is an important issue. In order to achieve a target bit rate, the quantizer step size is no longer fixed, but determined by the rate control algorithm. Furthermore, although a target frame-rate is generally specified, the rate control algorithm is free to drop individual source pictures when the bit-rate budget is exceeded.

III.2.2.1 Low complexity coding model

The low complexity model includes all the mechanism described in III.2.1.1, with the addition of rate control to achieve the target bit rate. Details can be found in III.3.1.2 (low complexity motion vector search), III.4.1.2 (low complexity mode decision), III.3.4 (advanced INTRA coding mode), III.3.5 (modified quantization mode), and III.4.2 (rate control).

III.2.2.2 High quality coding model

The high quality model includes all the mechanisms described in III.2.1.2, with the addition of rate control to achieve the target bit rate. In this appendix, the use of rate control when combined with the high quality motion estimation and mode decision algorithms employs some simplifications, as described in III.4.2.3.

III.2.3 Fixed bit-rate coding for transmission over packet-switched lossy channels (H.323)

This application scenario is characterized by the need to achieve a fixed target bit rate, and a packetization method for transport. In H.323 systems, an RTP-based packet oriented transport is used [RFC 1889]. The main characteristics of such a transport are as follows:

- variable packet sizes as determined by the sender, in the neighborhood of 1.5 kbyte, to reduce packetization overhead and to match the Internet maximum transfer unit (MTU) size; and
- significant packet loss rates.

Note that mid-term averages of packet loss rates are available to the encoder from mandatory RTCP receiver reports, which are part of RTP. Since RTP and the associated underlying protocol layers ensure that packets are delivered in the correct sequence[1] and that they are bit error-free, the only source of error is packet loss.

It is well-known that bidirectional communication employing back-channel messages can greatly improve the reproduced picture quality. However, bidirectional communication is often not feasible due to the possible multicast nature of the transport and application, and due to the possible transmission delay constraints of the application. Also the complexity of simulating a truly bidirectional environment is non-trivial. For these reasons, only unidirectional communication is considered in the following.

---

[1] While RTP does not ensure correct sequence numbering as a protocol functionality, its header includes a sequence number that can be used to ensure correct sequencing of packets.

First, a fixed bit rate control algorithm is employed, to simplify the interaction between source rate control and transport mechanisms. In real systems, while RTP's buffer management may smooth short-term variations in transmission rates, the target bit rate and the receiver buffer size are typically adjusted periodically based on factors such as the average packet loss rate and billing constraints. Second, constant short-term average packet loss rates are assumed, to simplify the interaction of error resilience and transport mechanisms. In real systems, error resilience support should be adaptive, e.g. based on windowed average packet loss rates.

The remainder of this clause discusses packetization and depacketization issues and the application of video coding tools for this scenario. There are two different models, corresponding to low complexity and high quality.

III.2.3.1 Packetization and depacketization

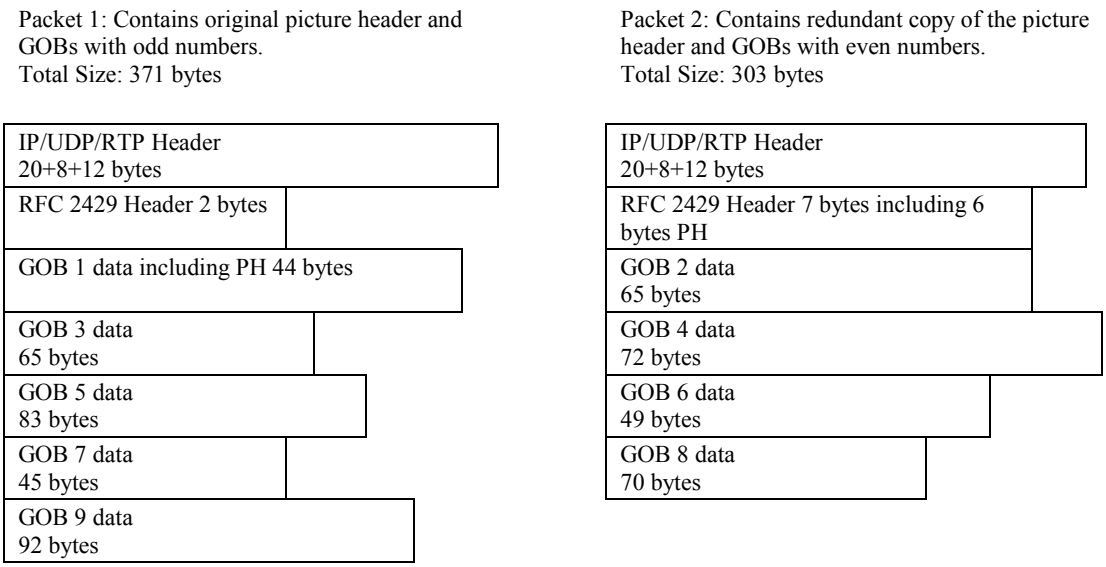
This subclause describes a packetization/depacketization scheme using RFC 2429 [RFC 2429] as the payload format for RTP that has been demonstrated to perform well for the environment assumed in this scenario.

The use of an encoding mechanism to reduce temporal error propagation, which is inevitable in a packet-lossy environment, is assumed. One such mechanism, described in III.4.1.1, is judicious use of INTRA coded macroblocks. The use of the Slice Structured mode (Annex K), where the slices are of the same size and shape as a GOB, is also assumed. In this scenario, slices are used instead of GOBs because Annex K permits a bitstream structure whereby the slices need not appear in the regular scan-order. The packetization scheme depends on such an arrangement.

The packetization scheme is based on interleaving even and odd numbered GOB-shaped slices, and arises from two design considerations. First, since the packetization overhead for the IP/UDP/RTP headers on the Internet is approximately 40 bytes per packet, reasonably big packets have to be used. Second, to allow for effective error concealment, as described in III.5.4, consecutive slices should not be placed in the same packet. This results in two packets per picture and allows for reasonable concealment of missing macroblocks if only one of the two packets is lost. This method may be extended to more than two packets per picture if the coded picture size is larger than 2800 bytes, where a maximum payload size of 1400 bytes per packet and an MTU size of 1500 bytes are assumed.

The loss of the contents of the picture header can seriously degrade decoded picture quality. This may be concealed by assuming that the contents of the picture header are unchanged, except for the temporal reference field, which can be recovered from the time reference included in the RTP header. When the video encoder changes picture header content, an RFC 2429 mechanism whereby redundant copies of the picture header can be included into the payload header of each packet is employed to allow for partial decoding of a picture (and subsequent concealment) on receipt of a single packet.

A typical example of this packetization scheme for a video bitstream coded at 50 kbit/s and 10 frames per second at QCIF resolution is presented in Figure III.1. The constant packetization overhead (consisting of the IP/UDP/RTP headers, 40 bytes per packet in total) is 80 bytes per packetized picture. The term GOB in the figure relates to a GOB-shaped slice whose spatial position corresponds to the GOB with the associated number.



PH H.263 Picture Header

Figure III.1/H.263 – An example of two packets of a picture using the interleaved packetization scheme

The 2 bytes minimum header defined in RFC 2429 does not contribute to the overhead, because this header replaces the 16-bit Picture Start Code or Slice Start Code that proceeds each picture/slice in H.263.

For the above packetization scheme, in a packet-lossy environment, four different situations can occur during depacketization and reconstruction. In the case that both packets are received, decoding is straightforward. In the case that only the first packet is received, the available data is decoded and the missing slices are detected and concealed for example as described in III.5.4. In the case that only the second packet is available, the redundant picture header and the payload data are concatenated and decoded. Missing slices are detected, as they cause syntax violations, and concealed as described in III.5.4. If both packets are lost, no data is available for the decoder and re-displaying the previous picture must conceal the entire picture.

III.2.3.2 Low complexity coding model

The low complexity coding model uses the same encoder mechanisms as described in III.2.2.1 with two additions that improve error resilience. GOB-shaped slices are employed, with slice headers inserted at the beginning of each slice. These serve as synchronization markers and to reset spatial predictive coding, such as motion vector coding and inter-coefficient coding, as defined in Annex K of H.263. Also, the rate at which macroblocks are forced to be coded in INTRA mode (refreshed) is varied with the inverse of the mid-term average packet loss rate. This is implemented via the INTRA_MB_Refresh_Rate of the INTRA macroblock refresh mechanism, as described in III.4.1.1.

III.2.3.3 High quality coding model

The high coding model uses the packetization, error concealment and GOB-shaped slices described above. Further, an extension of the high quality mode as described in III.2.1.2, whereby the high quality coding mode decision of III.4.1.3 is replaced by the high quality coding mode decision of III.4.1.5. This mode decision is motivated by the error-prone nature of the network, the packetization process, the packet loss rate, and the error concealment employed by the decoder to select the macroblock coding mode. Details can be found in III.4.1.5 (high quality mode decision for error-prone environments).

III.2.4 Fixed bit-rate coding for transmission over highly bit-error-prone channels (Annex C/H.324)

This application scenario is characterized by the need to achieve a fixed target bit rate in a highly bit-error-prone environment. In such environments H.223, the transport protocol employed by the corresponding Annex C/H.324 system, is primarily optimized for low delay operation. H.223 cannot provide bit error-free delivery of the payload, even when using the optional retransmission algorithms. Therefore, the video decoder must be able to detect and handle bit errors.

For practical reasons, several assumptions and simplifications regarding the transport simulation are necessary:

- the framed mode of H.223 is employed, with AL3 SDUs for H.263 data (allowing for the 16-bit CRC to detect errors);
- unidirectional communication is assumed, i.e. no retransmission algorithms or back-channel mechanisms are used, due to the stringent delay constraints.

Two models are defined: a low complexity coding model, that relies only on coding tools available in Version 2 (1998) of H.263, and a high error resilience coding model, that includes tools available in Version 3 of H.263.

III.2.4.1 Low complexity coding model

The low complexity model uses the same video coding tools as the fixed bit rate, error-free environment low complexity model, described in III.2.2.1. To improve error resilience, a packetization method and a forced INTRA coding method are also included. Each GOB is coded with a GOB header. This header serves as a synchronization marker and resets spatial predictive coding, such as motion vector coding and inter-coefficient coding, as defined in H.263. Each coded GOB is packetized into one AL3 SDU. Any received AL3 SDUs that fails the CRC test are not processed by the decoder, but concealed. No attempt is made to decode partially corrupted GOBs due to the difficulty to exactly define a decoder's operation in such a case. Further, this problem is compounded by the fact that errors which are not detected may result in very visible artefacts such as differently coloured blocks. The only syntax violation that is allowed, and used to detect missing GOBs at the decoder, is the out-of-sequence GOB numbering.

A fixed INTRA macroblock refresh rate is used. To determine the INTRA macroblock refresh rate, a rule of thumb is employed as follows. Every $1/p$'th time (rounded to the closest integer value) a given macroblock is coded containing coefficients, it is to be coded in INTRA mode, where p is the average loss probability for all macroblocks of a sequence for a given error characteristic. That is, if the determined loss probability is 0.1, then every 10th time a macroblock containing coefficient information is to be coded in INTRA mode. This algorithm is to be implemented via the INTRA_MB_Refresh_Rate of the INTRA macroblock refresh mechanism, as described in III.4.1.1. In contrast to the algorithm recommended in 5.1.4.3 of H.263 Version 2, the Rounding Type bit of PLUSPTYPE is set to "0" regardless of the picture type [Q15-I-26].

III.2.4.2 High error resilience coding model

The high error resilience coding model relies on the Data Partitioned Slice mode, as discussed in III.4.2.4. Furthermore, the "previous picture header repetition" mechanism of Annex W is employed as discussed in III.4.2.6.

III.3 Common algorithms

This clause discusses algorithms common to all application scenarios.

III.3.1 Motion estimation

In this appendix, three motion estimation algorithms are described:

- a low complexity, fast-search algorithm based on a reduced number of search locations;
- a medium complexity, full-search algorithm; and
- a high complexity, full-search algorithm that considers the resulting motion vector bit rate in addition to the quality of the match.

Of these, the low and high quality algorithms are the most frequently used, for the corresponding low and high quality models of the application scenarios. The medium quality algorithm can replace the low complexity algorithm and provides slightly better rate-distortion performance at much higher computational complexity. Note that any block-based search algorithm can be employed. However, the three described in the Test Model have been widely used and are known to perform well for the various application scenario models.

H.263 can use one or four motion vectors per macroblock depending of the optional modes that are enabled. Please refer to Annexes D, F, and J of H.263 for a description of the various limitations of the motion vectors. Also, the permissible extent of the search range depends on the sub-mode of Annex D being employed.

III.3.1.1 Sum of absolute difference distortion measure

Both integer-pixel and half-pixel motion vector search algorithms use the sum of absolute difference (SAD) as a distortion measure. The SAD is calculated between all luminance-pixels of the candidate and target macroblocks. In some cases, the mode decision algorithm will favour the SAD for the (0,0) vector. The SAD for a candidate motion vector is calculated as follows:

$$SAD(u,v) = \sum_{j=0}^{M-1} \sum_{i=0}^{M-1} \left| \widetilde{X}_{t-1}(i+u, j+v) \right|$$

where:

$$M = \begin{cases} 8 & \text{for } 8\times 8 \text{ motion vectors} \\ 16 & \text{for } 16\times 16 \text{ motion vectors} \end{cases}$$

X_t the target frame

$\widetilde{X}_{t-1}$ the previous reconstructed frame

(i, j) the spatial location within the target frame

(u, v) the candidate motion vector

The SAD calculation employs the partial distortion technique to reduce complexity. The partial distortion technique compares the accumulated SAD after each row of M pixels to the minimum SAD found to date within the search window. If the accumulated SAD exceeds the minimum SAD found to date, the calculation for the candidate motion vector is terminated, as the candidate motion vector will not produce a better match, in terms of a lower SAD value, than the best match found so far.

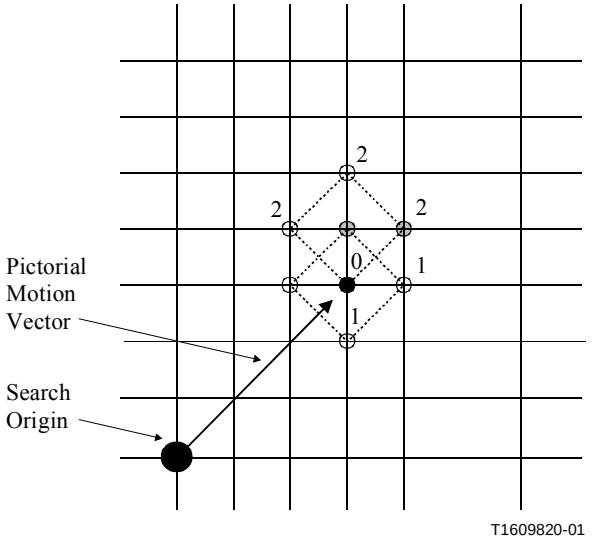
III.3.1.2 Low complexity search

The low complexity search was approved after extensive comparison to the full-search algorithm showed that little or no performance degradation was incurred for QCIF and CIF resolution pictures. Details are available in [GCK99], and extensive results comparing the performance to the full-search algorithm are available in [Q15-B-23].

III.3.1.2.1 Search procedure for 16 × 16 blocks and integer-pixel accuracy

The search centre is the median predicted motion vector as defined in 6.1.1 and F.2 of H.263. The (0,0) vector, if different than the predicted motion vector, is also searched. The (0,0) vector is favoured by subtracting 100 from the calculated SAD. The allowable search range is determined by the sub-mode of Annex D being employed.

The algorithm proceeds by sequentially searching diamond-shaped layers, each of which contains the four immediate neighbours of the current search centre. Layer $i + 1$ is then centred at the point of minimum SAD of layer i. Thus successive layers have different centres and contain at most three untested candidate motion vectors, except for the first layer around the predicted motion vector, which contains four untested candidate motion vectors. An example of this search is illustrated in Figure III.2.



0 Layer 0 search location (predicted motion vector)
1 Layer 1 search location
2 Layer 2 search location

NOTE – A shaded point represents the minimum SAD of the respective layer.

Figure III.2/H.263 – An example of the integer-pixel fast search extending two layers from the predicted motion vector

The search is stopped only after:

1) all candidate motion vectors in the current layer have been considered and the minimum SAD value of the current layer is larger than that of the previous layer; or
2) after the search reaches the boundary of the allowable search region and attempts to go beyond this boundary.

The resulting 16 × 16 integer-pixel MV is then half-pixel refined as described next.

III.3.1.2.2 16 × 16 half-pixel refinement

The half-pixel search area is ±1 half-pixel around the best 16 × 16 candidate. The search is performed by calculating half-pixel values as described in 6.1.2/H.263, and then calculating the SAD for each possible half-pixel vector. Thus, eight additional candidate vectors are tested. The vector resulting in the best match during the half-pixel refinement is retained.

III.3.1.2.3 8×8 integer-pixel search

The 8×8 integer motion vectors are set to the best 16×16 integer-pixel motion vector. There is no additional 8×8 integer-pixel search. Because of the differential encoding of motion vectors, this restriction ensures that the 8×8 motion vectors selected for the target macroblock are highly correlated.

III.3.1.2.4 8×8 half-pixel refinement

The half-pixel refinement is performed for each of the blocks around the 8×8 integer-pixel vector. The search is essentially the same as that described in III.3.1.2.2, with block sizes equal to 8. Note that the (0,0) vectors are not favoured here.

III.3.1.3 Medium quality, full search

III.3.1.3.1 16×16 integer-pixel search

The search centre is again the median predicted motion vector as defined in 6.1.1 and F.2 of H.263. The (0,0) vector, if different from the predicted motion vector, is also searched. The (0,0) vector is favoured by subtracting 100 from the calculated SAD. The allowable search range is determined by the sub-mode of Annex D being employed.

The algorithm proceeds by sequentially searching layers in an outward spiral pattern, with respect to the predicted motion vector, to the full extent of the permitted search range, for both the 16×16 and the 8×8 integer-pixel search. Thus each layer, except for the layer consisting of only the search origin, adds *8*layer_number* candidate motion vectors. For each layer, the search begins at the upper left corner of the spiral and proceeds in the clockwise direction.

This spiral search is more efficient if it originates from the median predicted motion vector, as this will usually produce a good match early in the search which, when combined with the partial distortion technique, can significantly reduce the number of row SADs that are calculated. Half-pixel refinement is identical to the low complexity algorithm, as described in III.3.1.2.2.

III.3.1.3.2 8×8 integer-pixel search

The 8×8 integer per search exhaustively searches a reduced search window, in the neighbourhood the best 16×16 integer-pixel motion vector, of size ± 2 integer-pixel units. This permits a slightly greater variation in the 8×8 motion vectors for the target macroblock, but these vectors can now provide higher quality matches. Half-pixel refinement is identical to the low complexity algorithm, as described in III.3.1.2.4.

III.3.1.4 High quality, rate-distortion optimized full search

In order to regularize the motion estimation problem, a Lagrangian formulation is used wherein distortion is weighted against rate using a Lagrange multiplier. If any mode which supports 8×8 motion vectors is enabled, e.g. Annex F and/or Annex J, a full search is employed for both the 16×16 and 8×8 integer-pixel motion vectors. Otherwise, a full search is employed only for the 16×16 integer-pixel motion vectors.

For each 16×16 macroblock or 8×8 block, the integer-pixel motion vector search employs spiral search pattern, as described in III.3.1.3.1. The half-pixel refinement uses the pattern described in III.3.1.2.2. The half-pixel motion vector for 8×8 block i is obtained before proceeding to obtain the integer or half-pixel motion vectors for 8×8 block $i + 1$, etc. This allows for accurate calculation of the rate terms for the integer and half-pixel motion vectors for the 8×8 blocks, as the predicted motion vector will be completely known.

The RD optimized integer-pixel search selects the motion vector that minimizes the Lagrangian cost defined as:

$$J = D + \lambda_{motion} R$$

The distortion, *D*, is defined as the SAD between the luminance component of the target macroblock (or block) and the macroblock (or block) in the reference picture displaced by the candidate motion vector. The SAD calculations use the partial distortion matching criteria using the minimum SAD found to date. The rate, *R*, is defined as the sum of the rate for the vertical and horizontal macroblock (or block) motion vector candidates, taking into account the predicted motion vector as defined in 6.1.1/H.263. The parameter λ is selected as described below. The search is centred at the predicted motion vector. The (0,0) vector is searched but not favoured, as the Lagrangian minimization already accounts for the motion vector rate. The best 16×16 and 8×8 candidates are then half-pixel refined using a ± 1 half-pixel search. The half-pixel search also employs the same Lagrangian formulation as above, to select the half-pixel motion vector that minimizes the Lagrangian cost. The mode decision algorithm then uses the best 16×16 and 8×8 vectors.

The choice of λ_{motion} has a rather small impact on the result of the 16×16 block motion estimation. But the search result for 8×8 blocks is strongly affected by λ_{motion}, which is chosen as:

$$\lambda_{motion} = 0.92 \cdot QP$$

where *QP* is the macroblock quantization parameter [Q15-D-13]. If the application is for a fixed bit rate, the relationship is modified as described in III.4.2.3.

III.3.1.5 Other motion vector search related issues

Although not used in any application scenarios, the test model includes motion vector search algorithms for PB-frames (when Annex G or M are enabled), when in Reduced Resolution Update mode, and for the various picture types permitted with scalability mode defined in Annex O. These are discussed in more detail in III.6.1.1, III.6.3 and III.6.2.2 respectively.

III.3.2 Quantization

The quantization parameter *QUANT* may take integer values from 1 to 31. The quantization reconstruction spacing for non-zero coefficients is $2 \cdot QP$, where:

$QP = 4$	for INTRA DC coefficients when not in Advanced INTRA Coding mode; and
$QP = QUANT$	otherwise.

Define the following:

COF	A transform coefficient (or coefficient difference) to be quantized
LEVEL	The quantized version of the transform coefficient
REC	Reconstructed coefficient value
"/"	Division by truncation

The basic inverse quantization reconstruction rule for all non-zero quantized coefficients can be expressed as:

$$|REC| = QP \cdot (2 \cdot |LEVEL| + p) \quad \text{if } QP = \text{"odd"; and}$$

$$|REC| = QP \cdot (2 \cdot |LEVEL| + p) - p \quad \text{if } QP = \text{"even",}$$

where:

$p = 1$ for INTER coefficients;

$p = 1$ for INTRA non-DC coefficients when not in Advanced INTRA Coding mode;

$p = 0$ for INTRA DC coefficients when not in Advanced INTRA Coding mode; and

$p = 0$ for INTRA coefficients (DC and non-DC) when in Advanced INTRA Coding mode.

The parameter p is unity when the reconstruction value spacing is non-uniform (i.e. when there is an expansion of the reconstruction spacing around zero), and p is zero otherwise. The encoder quantization rule to be applied is compensated for the effect that p has on the reconstruction spacing. In order for the quantization to be MSE-optimal, the quantizing decision thresholds should be spaced so that the reconstruction values form an expected-value centroid for each region. If the probability density function (pdf) of the coefficients is modelled by the Laplacian distribution, a simple offset that is the same for each quantization interval can achieve this optimal spacing. The coefficients are quantized according to such a rule, i.e. they use an "integerized" form of:

$$|LEVEL| = \left[|COF| + (f - p) \cdot QP \right] / (2 \cdot QP)$$

where:

$f \in \left\{ \frac{1}{2}, \frac{3}{4}, 1 \right\}$ is a parameter that is used to locate the quantizer decision thresholds such that each reconstruction value lies somewhere between an upward-rounding nearest-integer operation ($f = 1$) and a left-edge reconstruction operation ($f = 0$), and f is chosen to match the average (exponential) rate of decay of the pdf of the source over each non-zero step.

III.3.2.1 Quantization for INTER coefficients

INTER coefficients (whether DC or not) are quantized according to:

$$|LEVEL| = \left(|COF| - QUANT / 2 \right) / \left(2 \cdot QUANT \right)$$

This corresponds to $f = \frac{1}{2}$ with $p = 1$.

III.3.2.2 Quantization for INTRA non-DC coefficients when not in advanced INTRA coding mode

INTRA non-DC coefficients when not in Advanced INTRA Coding mode are quantized according to:

$$|LEVEL| = |COF| / \left(2 \cdot QUANT \right)$$

This corresponds to $f = 1$ with $p = 1$.

III.3.2.3 Quantization for INTRA DC coefficients when not in advanced INTRA coding mode

The DC coefficient of an INTRA block when not in Advanced INTRA Coding mode is quantized according to:

$$LEVEL = \left(COF + 4 \right) / \left(2 \cdot 4 \right)$$

This corresponds to $f = 1$ with $p = 0$. Note that *COF* and *LEVEL* are always non-negative and that *QP* is always 4 in this case.

III.3.2.4 Quantization for INTRA coefficients when in advanced INTRA coding mode

INTRA coefficients when in Advanced INTRA Coding mode (DC and non-DC) are quantized according to:

$$|LEVEL| = \left(|COF| + 3 \cdot QUANT / 4\right) / \left(2 \cdot QUANT\right)$$

This corresponds to $f = \frac{3}{4}$ with $p = 0$.

III.3.3 Alternative INTER VLC mode (Annex S)

During the entropy coding the encoder will use the Annex I INTRA VLC table for coding an INTER block if the following three criteria are satisfied:

- Annex S, alternative INTER VLC mode is used and signalled in the picture header;
- coding the coefficients of an INTER block using the Annex I INTRA VLC table results in fewer bits than using the INTER VLC table;
- the use of Annex I INTRA VLC table shall be detectable by the decoder. The decoder assumes that the coefficients are coded with the INTER VLC table. The decoder detects the use of Annex I INTRA VLC table when coefficients outside the range of 64 coefficients of an 8×8 block are addressed.

With many large coefficients, this will easily happen due to the way the INTRA VLC was designed, and a significant number of bits can be saved at high bit rates.

III.3.4 Advanced INTRA coding mode (Annex I)

Advanced INTRA coding is a method to improve intra-block coding by using inter-block prediction. The application of this technique is described in Annex I of H.263. The procedure is essentially intra-block prediction followed by quantization as described in III.3.2.4, and the use of different scanning orders and a different VLC table for entropy coding of the quantized coefficients.

Coding for intra-blocks is implemented by choosing one among the three predictive modes described in H.263. Note that H.263 employs the *reconstructed* DCT coefficients to perform the inter-block prediction, whereas the *original* DCT coefficients are employed in the encoder to decide on the prediction mode. This Test Model describes the mode decision performed at the encoder based on the original DCT coefficients.

The blocks of DCT coefficients employed during the prediction are labelled A(u, v), B(u, v) and C(u, v), where u and v are row and column indices, respectively. C(u, v) denotes the DCT coefficients of the block to be coded, A(u, v) denotes the block of DCT coefficients immediately above C(u, v) and B(u, v) denotes the block of DCT coefficients immediately to the left of C(u, v). The ability to use the reconstructed coefficient values for blocks A and B in the prediction of the coefficient values for block C depends on whether blocks A and B are in the same video picture segment as block C. A picture segment is defined in Annex R of H.263. Ei(u, v) denotes the prediction error for intra mode i = 0, 1, 2. The prediction errors are computed for all three coding modes as follows:

```
mode 0: DC prediction only.
If (block A and block B are both intra coded and are both in the same
picture segment as block C)
{
 E0(0,0) = C(0,0) – ( A(0,0) + B(0,0) )//2
}
else
{
 If (block A is intra coded and is in the same picture segment as block C)
 {
```

```
  E0(0,0) = C(0,0) – A(0,0)
 }
 else
 {
  If (block B is intra coded and is in the same picture segment as block C)
  {
   E0(0,0) = C(0,0) – B(0,0)
  }
  else
  {
   E0(0,0) = C(0,0) – 1024
  }
 }
}

E0(u,v) = C(u,v)    u!=0, v!=0, u = 0..7, v = 0..7.

mode 1: DC and AC prediction from the block above.
If (block A is intra coded and is in the same picture segment as block C)
{
 E1(0,v) = C(0,v) – A(0,v)     v = 0..7, and
 E1(u,v) = C(u,v)         u = 1..7, v = 0..7.

}
else
{
 E1(0,0) = C(0,0) – 1024
 E1(u,v) = C(u,v)      (u,v) != (0,0), u = 0,_,7, v = 0,_,7
}

mode 2: DC and AC prediction from the block to the left.
If (block B is intra coded and is in the same picture segment as block C)
{
 E2(0,v) = C(u,0) – A(u,0)    u = 0..7, and
 E2(u,v) = C(u,v)        v = 1..7, u = 0..7.

}
else
{
 E2(0,0) = C(0,0) – 1024
 E2(u,v) = C(u,v)      (u,v) != (0,0), u = 0,_,7, v = 0,_,7
}
```

The prediction mode selection for advanced INTRA coding is done by evaluating the absolute sum of the prediction error, $SAD_{mode\ i}$, for the four luminance blocks in the macroblock and selecting the mode with the minimum value.

$$\mathrm{SAD}_{\mathrm{mode}\,i} = \sum_{b}\left[\left|E_i(0,0)\right| + 32\sum_{u}\left|E_i(u,0)\right| + 32\sum_{v}\left|E_i(0,v)\right|\right]$$

where:

$i =$ 0 ... 3,

$b =$ 0 ... 3,

$u, v =$ 1 ... 7.

Once the appropriate mode is selected, quantization is performed. The blocks are quantized as described in III.3.2.4.

III.3.5 Modified quantization mode (Annex T)

Annex T, Modified Quantization, greatly reduces certain colour artefacts (particularly at low bit rates) and increases the range of luminance coefficients. Moreover, Annex T allows the encoder to set the quantizer step size to any value at the macroblock granularity, which may improve the performance of rate control algorithms. Annex T is mandated for all application scenarios and is strongly encouraged for low-bit-rate product designs.

III.4 Algorithms used for individual application scenarios

This clause discusses algorithms used for specific application scenarios.

III.4.1 Mode decision

H.263 allows for several types of macroblock coding schemes, such as INTRA mode (coding non-predicted DCT coefficients), INTER mode (predictive coding using 1 motion vector) and INTER4V (predictive coding using 4 motion vectors). The choice of the appropriate mode is one of the key functionalities of an encoder and the quality of the decision influences greatly the performance of the encoder. A high quality and a low complexity algorithm for both error-free and error-prone environments have been developed and are described in the following subclauses. First, a mechanism for performing the mandatory INTRA update is described.

III.4.1.1 INTRA macroblock refresh and update pattern

As required by H.263, every macroblock must be coded in INTRA mode at least once every 132 times when coefficients are transmitted (unless the Annex W IDCT is in use). To avoid large bursts of INTRA macroblocks for short periods, a simple pattern for the macroblock update is used to randomly initialize each macroblock's update counter to a value in the range [0,132]. The pseudo-random generator of H.263 Annex A should be used to initialize the random pattern. This mode decision process overrides the coding mode after any other mode decision process. The algorithm is described as follows:

```
MB_intra_update[xpos][ypos]: counter incremented by one every time
coefficient information is sent for the macroblock at position (xpos, ypos),
namely MB[xpos][ypos].

INTRA_MB_Refresh_Rate: in error-free environments a constant (132). In
error-prone environments, INTRA_MB_Refresh_Rate might be an integer variable
(adapted to the error rate), with a value in (1, 132).
```

The INTRA mode for a given macroblock is chosen as:

```
Initialize after an I-picture: MB_intra_update[xpos][ypos] =
random_of_Annex_A (0, INTRA_MB_Refresh_Rate)
 While (more non I-picture to encode)
  If ((MB_intra_update[xpos][ypos] == INTRA_MB_REFRESH_RATE) &&
      (MB[xpos][ypos] contains coefficient))
  {
   Encode MB[xpos][ypos] in INTRA mode;
  }
  else if (MB[xpos][ypos] contains coefficient)
  {
   ++MB_intra_update[xpos][ypos];
  }
```

Further details are available in [Q15-E-15] and [Q15-E-37].

III.4.1.2 Low complexity mode decision for error-free environments

Figure III.3 outlines the low complexity mode decision algorithm, which interacts with the steps of the low and medium quality motion vector search, described in III.3.1.2 and III.3.1.3, respectively.

After performing integer-pixel motion estimation, the encoder selects the prediction mode, INTRA or INTER. The following parameters are calculated to make the INTRA/INTER decision:

$$\mu_{MB} = \frac{1}{256}\sum_{i=1}^{16}\sum_{j=1}^{16} X(i, j)$$

$$A = \sum_{i=1}^{16}\sum_{j=1}^{16} \left| X(i, j) - \mu_{MB} \right|$$

INTRA mode is chosen if:

$$A < \left(SAD(u, v) - 500 \right)$$

If INTRA mode is chosen, no further mode decision or motion estimation steps are necessary. If rate control is in use, the DCT coefficients of the blocks are quantized using quantization parameter determined by rate control. Advanced INTRA coding mode operations are performed as defined in III.3.4. The coded block pattern (CBP) is calculated and the bitstream for this macroblock is generated.

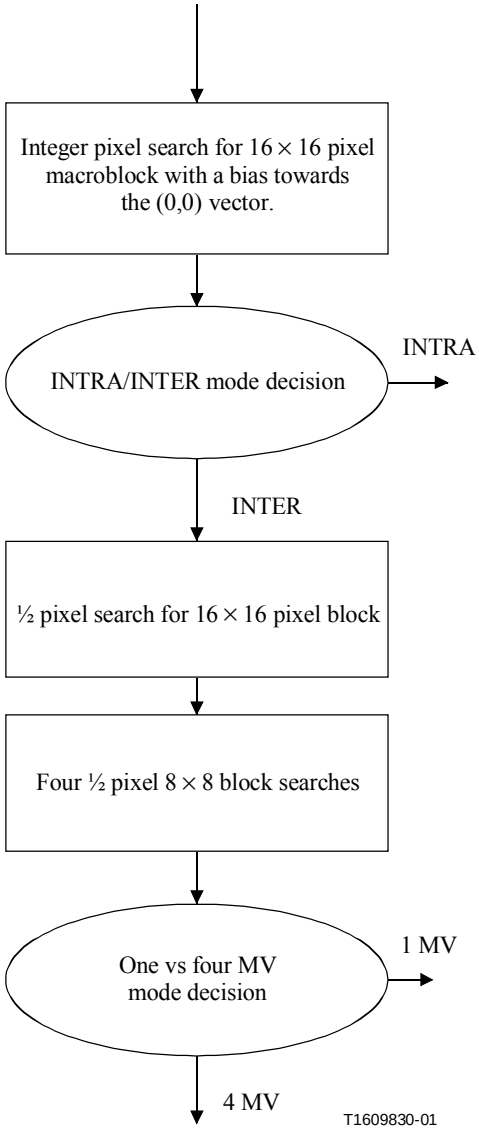


Figure III.3/H.263 – Block diagram of the low complexity mode decision process

If INTER mode is chosen the motion search continues with a half-pixel refinement around the 16×16 integer-pixel MV.

If 8×8 motion vectors are permitted, i.e. either Annex F or Annex J, or both are enabled, the motion search also continues with half-pixel search around the 8×8 integer-pixel MVs. The mode decision then proceeds to determine whether to use INTER or INTER4V prediction.

The SAD for the half-pixel accuracy 16×16 vector (including subtraction of 100 if the vector is (0,0)) is:

$$\mathrm{SAD}_{16}(u,v)$$

The aggregate SAD for the 4 half-pixel accuracy 4 half 8×8 vectors is:

$$\mathrm{SAD}_{8\times 8} = \sum_{k=1}^{4} \mathrm{SAD}_{8,k}(u_k, v_k)$$

INTER4V mode is chosen if:

$$\mathrm{SAD}_{8\times 8} < \mathrm{SAD}_{16} - 200$$

otherwise INTER mode is chosen.

III.4.1.3 High quality mode decision for error-free environments

High quality motion estimation for the INTER and INTER4V mode is conducted first, as described in III.3.1.4. Given these motion vectors, the overall rate-distortion costs for all considered coding modes are computed.

Independently, the encoder selects the macroblock coding mode that minimizes:

$$J = D + \lambda_{\text{mode}} \times R$$

The distortion, D, is defined as the sum of squared difference (SSD) between the luminance and chrominance component coefficients of the target macroblock and the quantized luminance and chrominance component coefficients for the given coding mode. The rate R is defined as the rate to encode the target macroblock including all control, motion, and texture information, for the given coding mode.

The parameter λ is selected as:

$$\lambda_{\text{mode}} = 0.85 \cdot QP^2$$

where QP is the macroblock quantization parameter [Q15-D-13].

The coding modes explicitly tested are SKIPPED, INTRA, INTER, and INTER4V. The INTER mode uses the best 16×16 motion vector and the INTER4V mode uses the best 8×8 motion vectors as selected by high quality motion estimation algorithm. If the application is for a fixed bit rate, the relationship is modified as described in III.4.2.3.

If Annex F is in use, the overlapped block motion compensation window is not considered for making coding mode decisions. It is applied only after all motion vectors have been determined and mode decisions have been made for the picture.

III.4.1.4 Low complexity mode decision for error-prone environments

The low complexity mode decision algorithm for error-prone environments uses the same mechanism as described in III.4.1.2, with one addition. The INTRA macroblock update frequency is increased, by using a value less than 132 for the INTRA_MB_Refresh_Rate parameter of III.4.1.1. Guidelines for choosing a value for this variable can be found in III.2.3.2 and III.2.4.1.

III.4.1.5 High quality mode decision for error-prone environments

The high quality mode decision algorithm for error-prone environment is similar to the one described in III.4.1.3. Independently for every macroblock, the encoder selects the coding mode that minimizes:

$$J = (1-p)D_1 + pD_2 + \lambda_{\text{mode}} R$$

where λ_{mode} is the same as III.4.1.3. To compute this cost, the encoder simulates the error concealment method employed by the decoder and must obtain the probability that a given macroblock is lost with probability p during transmission. The distortion thus arises from two sources, where D_1 is the distortion attributed to a received macroblock, and D_2 is the distortion attributed to a lost and concealed macroblock.

For a given MB, two distortions are computed for all coding modes considered: the coding distortion D_1 and the concealment distortion D_2. D_1 is weighted by the probability $(1-p)$ that this MB is received without error, and D_2 is weighted by the probability p that the same MB is lost and concealed. D_2 depends on the error concealment method employed and is constant for all coding modes considered (and therefore has no effect on which mode is selected using the above equation).

D_1 will depend on the coding mode considered. For the INTRA mode, the distortion is the same as the error-free case. For the SKIP and INTER modes, D_1 is the quantization distortion plus the concealment distortion of the macroblock from which it is predicted, more precisely,

$$D_1 = D_q + pD_2(v, n-1)$$

where $D_2(v, n-1)$ represents the concealment distortion of the area of the previous picture, $(n-1)$, at the current spatial location of the macroblock displaced by the motion vector v, where v is set to *zero* for the SKIP mode and set to the motion vector of the considered macroblock for INTER mode.

D_q is defined as the SSD between the properly-reconstructed pixel values of the coded macroblock and the original pixel values, and D_2 is defined as the SSD between the properly-reconstructed pixel values of the coded macroblock and the concealed pixel values. The rate R is defined as the rate to encode the target macroblock including all control, motion, and texture information, for the given coding mode.

III.4.2 Rate control

This rate control algorithm applies for all fixed bit-rate application scenarios. It consists of a frame-layer, in which a target bit rate for the current picture is selected, and a macroblock-layer, in which the quantization parameter (QP) is adapted to achieve that target. The development of this algorithm is described in [Q15-A-20]. This rate control algorithm, often referred to as TMN8 rate control, is the only rate control algorithm used for the application scenarios.

Initially, the number of bits in the buffer W is set to zero, $W = 0$, and the parameters K_{prev} and C_{prev} are initialized to $K_{prev} = 0.5$ and $C_{prev} = 0$. The first picture is intra-coded using a fixed value of QP for all macroblocks.

III.4.2.1 Frame level rate control

The following definitions are used in describing the algorithm:

B' Number of bits occupied by the previous encoded picture.

R Target bit rate in bits per second (e.g. 10 000 bit/s, 24 000 fps, etc.).

G Frame rate of the original video sequence in pictures per second (e.g. 30 fps).

F Target frame rate in pictures per second (e.g. 7.5 fps, 10 fps, etc.). G/F must be an integer.

M Threshold for frame skipping. By default, set $M = R/F$. (M/R is the maximum buffer delay.)

A Target buffer delay is AM s. By default, set $A = 0.1$.

The number of bits in the encoder buffer is $W = \max(W + B' - R/F, 0)$. The skip parameter is set to 1, skip = 1.

```
While W > M
{
     W= max (W - R/F, 0)
     skip++
}
```

Skip encoding the next (skip*(G/F) – 1) pictures of the original video sequence. The target number of bits per picture is B = (R/F) – Δ, where:

$$\Delta = \begin{cases} \dfrac{W}{F}, & W > A \cdot M \\ W - A \cdot M, & \text{otherwise} \end{cases}$$

For fixed frame rate applications, skip is always equal to 1. The frame skip is constant and equal to (*G/F*) – 1. Also, when computing the target number of bits per picture, *A* should be set to 0.5.

III.4.2.2 Macroblock level rate control

Step 1: Initialization

It is assumed that the motion vector estimation has already been completed.

σ_k^2 is defined as the variance of the luminance and chrominance values in the kth macroblock.

If the kth macroblock is of type I (intra), set $\sigma_k^2 = \sigma_k^2 / 3$.

Let i = 1 and j = 0;

$\widetilde{B}_1 = B$ the target number of bits as defined in III.4.2.1;

$N_1 = N$ the number of macroblocks in a picture;

$K = K_1 = K_{prev}$ the initial value of the model parameters; and

$C = C_1 = C_{prev}$ the initial value of the model parameters.

$$S_1 = \sum_{k=1}^{N} \alpha_k \sigma_k \text{, where } \alpha_k = \begin{cases} 2\dfrac{B}{16^2 N}(1-\sigma_k)+\sigma_k, & \dfrac{B}{16^2 N} < 0.5, \\ 1, & \text{otherwise.} \end{cases}$$

Step 2: Compute optimized Q for i-th macroblock

If $L = (\widetilde{B}_i - 16^2 N_i C) \le 0$ (running out of bits), set $Q_i^* = 62$.

Otherwise, compute:

$$Q_i^* = \sqrt{\frac{16^2 K}{L}}\frac{\sigma_i}{\alpha_i} S_i$$

Step 3: Find QP and encode macroblock

QP = round $\left(Q_i^* / 2\right)$ to nearest integer in set 1, 2, …, 31.

DQUANT = QP – QP_prev.

If DQUANT > 2, set DQUANT = 2. If DQUANT < –2, set DQUANT = –2.

Set QP = QP_prev + DQUANT.

DCT encode macroblock with quantization parameter QP, and set QP_prev = QP.

Step 4: Update counters

Let $B_i^{'}$ be the number of bits used to encode the i-th macroblock, compute:

$$\widetilde{B}_{i+1} = \widetilde{B}_i - B_i^{'},\ S_{i+1} = S_i - \alpha_i \sigma_i, \text{ and } N_{i+1} = N_i - 1$$

Step 5: Update model parameters K and C

The model parameters measured for the i-th macroblock are:

$$\hat{K} = \frac{B_{LC,i}^{'}(2QP)^2}{16^2 \sigma_i^2} \text{ and } \hat{C} = \frac{B_i^{'} - B_{LC,i}^{'}}{16^2}$$

where $B'_{LC,i}$ is the number of bits spent for the luminance and chrominance of the macroblock.

Next, measure the average of the $\hat{K}$ s and $\hat{C}$ s computed so far in the picture.

If ($\hat{K} > 0$ and $\hat{K} \le \pi \log_2 e$), set j = j + 1 and compute $\widetilde{K}_j = \widetilde{K}_{j-1}(j-1)/j + \hat{K}/j$.

Compute $\widetilde{C}_i = \widetilde{C}_{i-1}(i-1)/i + \hat{C}/i$.

Finally, the updates are a weighted average of the initial estimates, K_1, C_1, and their current average:

$$K = \widetilde{K}_j(i/n) + K_1(N-i)/N, \qquad C = \widetilde{C}_i(i/n) + C_1(N-i)/N .$$

Step 6:

If i = N, stop (all macroblocks are encoded).

Set $K_{prev} = K$ and $C_{prev} = C$.

Otherwise, let i = i + 1, and go to Step 2.

III.4.2.3 High quality models and rate control

For Lagrangian-based minimizations, such as those described in the high quality motion estimation and mode decision algorithms in III.3.1.4 and III.4.1.3 respectively, the Lagrangian parameter must remain constant over an entire picture. This is because at optimality, all picture regions must operate at a constant slope point on their rate-distortion curves [SG88]. In theory, this optimal operating point is found via a search through all possible values of λ. However, in this test model, a simple relationship between λ and the quantization parameter is defined which produces a good approximation to the optimal value of λ [Q15-D-13].

Because the rate control algorithm described above can update the quantization parameter at each coded macroblock, the relationships between λ and the quantization parameter must be modified slightly. It has been shown that, to employ the high quality models for fixed bit-rate scenarios, the Lagrangian parameters should be calculated as:

$$\lambda_{\text{motion}} = 0.92 \cdot \overline{QP}_{prev} \text{ and}$$

$$\lambda_{\text{mode}} = 0.85 \cdot \overline{QP^2}_{prev}$$

respectively, where $\overline{QP}_{prev}$ is the average value, over all macroblocks, of the quantization parameter from the previous coded picture of the same type as the current picture.

The high quality motion estimation and mode decision algorithms then proceed to determine appropriate motion vectors and macroblock coding modes, as in the variable bit-rate scenarios, i.e. as if no rate control algorithm was enabled. Once the optimal motion vectors and modes have been determined, the rate control algorithm can then be employed to encode the picture based on the already determined motion vectors and coding modes. In the special case that the mode decision algorithm has selected *SKIPPED* mode for the target macroblock, the rate control algorithm is not permitted to update the quantization parameter.

III.4.2.4 Enhanced reference picture selection (Annex U)

The Enhanced Reference Picture Selection (ERPS) mode (Annex U) enables multi-picture motion-compensated prediction (MCP). In ERPS operation, M (M ≥ 1) prior decoded pictures are buffered at coder and decoder. In contrast to single-picture MCP, multi-picture MCP can exploit long-term statistical dependencies in video sequences. Typical examples for these dependencies are repeated scene cuts, covered and uncovered background, content fading in and out of the picture, etc.

III.4.2.4.1 Choice of the number of reference pictures

In real-world systems the number of reference pictures M that can be accommodated by the decoder is negotiated or otherwise established by external means. The rate-distortion improvements over single-picture MCP depend on the number of reference pictures and the manner in which they are used. More reference pictures will usually result in better coding performance. But an increased number of reference pictures also results in increased complexity and memory requirements. Averaging the results of several test sequences, it has been found that the improvements in PSNR at a fixed bit rate are typically proportional to the log(log(M)) of the number of reference pictures. Hence, for test model purposes, ten reference pictures are used to achieve a good trade-off between coding efficiency and complexity. Although an encoder may achieve enhanced performance by means of an optimized choice of which reference pictures are to be stored (and perhaps even an optimized choice of which sub-picture regions are to be stored), this test model simply stores and uses the reference pictures in a first-in, first-out (FIFO) manner.

III.4.2.4.2 Motion estimation

Motion estimation proceeds using the (up to M, which is 10 in this test model) reference pictures available in the multi-picture buffer. To determine the optimum motion vector $\boldsymbol{m}_k$ for the kth macroblock or block, the following Lagrangian cost function is minimized:

$$\boldsymbol{m}_k = \underset{\boldsymbol{m} \in \mathrm{SR}}{\operatorname{argmin}}\, D(\boldsymbol{s},\boldsymbol{c}(\boldsymbol{m})) + \lambda_{\mathrm{motion}}\, R(\boldsymbol{m} - \boldsymbol{p})$$

with SR being the search range typically comprising the set of integer-pixel positions $[-32...31] \times [-32...31] \times [0...M-1]$ in horizontal, vertical, and temporal direction.

NOTE – Most experiments conducted during proposal-phase testing of the ERPS mode actually used an integer-sample search range of ±16; however, a larger range is specified herein due to the larger range available in the test model design.

The distortion D($\boldsymbol{s}$, $\boldsymbol{c}(\boldsymbol{m})$) is given as SAD and the term R($\boldsymbol{m} - \boldsymbol{p}$) determines the rate including the spatial displacement m_x, m_y and the picture reference parameter m_t given the predictor $\mathbf{p} = (p_x, p_y, 0)$. After determination of the macroblock or block integer-pixel motion vector, the remaining encoding steps as specified in the high-quality mode of III.4.2.3 are applied. The Lagrange parameter $\lambda_{\mathrm{motion}}$ is chosen as $\lambda_{\mathrm{motion}} = 0.92\,\mathrm{QP}$.

III.4.2.5 Use of data partitioned slices (Annex V)

The data partitioned slice mode is used in the V3 mode of the highly bit error-prone application scenario. It is believed that it would also improve the performance in packet lossy environments as well, provided that the application of packetization schemes use at least one packet for each partition. By reorganizing the syntax and forming slices with three partitions, one each for header, motion vector, and DCT coefficients, the propagation of errors between partitions is reduced and RVLC usage enabled. In contrast to other parts of this test model, this clause includes algorithm descriptions for both encoder and decoder operation.

III.4.2.5.1 Encoder operation

The source encoding of the data is unchanged, only the syntax of the slices is changed from in Annex K. In order to map the coded video into the slices, a pseudo-fixed length packet structure is used. The encoder picks a slice length threshold t, which should be chosen based on network error and overhead characteristics. Once t has been chosen, the encoder codes an integral number of MB's n such that the resulting slice length is less than t. Ideally, t would result in slices with at most one or two errors in each of them to allow maximal advantage for RVLC recovery, and a good lower bound for packet size is therefore 0.5 * 1/BER bits, BER = average channel Bit Error Rate. However, this is not always possible due to the overhead involved in very short slices. With all of this in mind, the

test model implementation of Annex V has chosen t to be 700 bits, as this number was shown to produce good results for the 64 kbit/s, 10E – 3 BER WCDMA channel (which is the worst channel considered in the Common Conditions [Q15-I-60]).

III.4.2.5.2 Decoding method

This subclause describes the method used to decode the video bitstream including all employed error detection and concealment techniques. The informal algorithmic description overrides in so far the more general definitions of III.5.3. Error concealment is in general independent of the method described here and is therefore done in accordance with the method described in III.5.4.

As with the encoding method, the decoding of the actual video data is the same, the only difference is on data recovering and error handling given the new data-partitioned structure.

1) Scan the bitstream for a PSC. When we receive a PSC, we assume we have a picture header immediately following it.
2) Decode the picture header. This is exactly the same as when Annex V is not in use. If we detect an error in the picture header, we use the information from the last known error-free picture header or of Annex W's redundant copy, when available.
3) Scan for the next SSC. The next SSC should immediately follow the picture header and SSTUF. If not, we assume an error, and discard all bits between the picture header and this SSC.

   NOTE – Because the SSC is a prefix to the PSC, check to see if we indeed have a PSC. If so, the following repeat loop will of course not be executed.
4) Repeat the following until next PSC detected:
   a) Decode SEPB1 and SSBI.
   b) Immediately following SSBI is MBA. There are two cases:
      i) If this slice immediately follows a PSC and picture header, we know first MB index should be 0. If it is not 0, we correct it to 0 and continue decoding of the slice.
      ii) If not, we know what first MB index should be based on the immediately preceding slice. There are two cases:
         - no error in immediately preceding slice: set MBA to value predicted from immediately preceding slice;
         - missing or error in immediately preceding slice: set MBA to value read.
   c) Decode SEPB2, SQUANT, SWI, SEPB3, and GFID.
   d) Scan for next Header Marker (HM).
   e) Decode the header data (HD) partition. Decoding is only performed in the forward direction. If an error is detected, the whole slice is discarded until next PSC or SSC.
5) Scan for Motion Marker (MVM). If we find a PSC or SSC first, error is detected and the slice is discarded.
6) Decode the Motion Vectors partition (MVD and LMVV). This is done in the forward and backward direction. Retain all motion vectors decoded without error in both the forward and backward direction.
7) Scan for next PSC or SSC.
8) Decode the coefficient data partition. As it is VLC coded, not RVLC, we only decode in the forward direction. If an error is detected anywhere, the entire DCT coefficients are set to 0, the intuitive meaning of which is that the error residual information is lost.

III.4.2.6 Use of the previous header algorithm of Annex W

In contrast to the algorithm recommended in 5.1.4.3 of H.263 Version 2, the rounding type bit of PLUSPTYPE is set to "0" regardless of the picture type [Q15-I-26]. This helps in maintaining a constant value of GFID, which enables the decoder to recover a missing picture header as follows: If the GFID in the first decodable GOB header is the same as in the previous picture, the picture data is decoded using the picture header from the previous picture. Otherwise, if the GFID differs from the one in the previous picture, a copy of the picture header is inserted in the Supplemental Enhancement Information of the consecutive picture as defined in W.6.3.6 of H.263, and the decoder uses this copy to recover a missing picture header, see [Q15-J-62] and references therein.

III.5 Decoder post-processing

This clause defines the operation of the video decoder in error-prone environments, as well as post-processing methods such as error concealment and post-filtering algorithms. While post-filtering is useful in error-free and error-prone environments, all other mechanisms described here apply only to error-prone environments.

III.5.1 De-ringing post-filter

The de-ringing post-filter described in this subclause should be used whenever Annex J is in use. Otherwise, the de-blocking post-filter described in III.5.2 should be employed. For the definition of the functions and notation used in this subclause, see the definition of the loop filter in Annex J of H.263.

The one-dimensional version of the filter will be described. To obtain a two-dimensional effect, the filter is first used in the horizontal direction and then in the vertical direction. The pixels A, B, C, D, E, F, G, (H) are aligned horizontally or vertically. A new value – D1 – for D will be produced by the filter:

D1 = D + Filter ((A + B + C + E + F + G – 6D)/8, Strength1) when filtering in the first direction.

D1 = D + Filter ((A + B + C + E + F + G – 6D)/8, Strength2) when filtering in the second direction.

As opposed to Annex J filter, the post filter is applied to all pixels within the picture. Edge pixels should be repeated when the filter is applied at picture boundaries. Strength1 and Strength2 may be different to better adapt the total filter strength to QUANT. The relation between Strength1, 2 and QUANT is given in Table III.1. Strength1, 2 may be related to QUANT for the macroblock where D belongs or to some average value of QUANT over parts of the picture or over the whole picture.

A sliding window technique may be used to obtain the sum of seven pixels (A + B + C + D + E + F + G). In this way the number of operations to implement the filter may be reduced.

Table III.1/H.263 – TMN

QUANT	Strength	Strength1	Strength2	QUANT	Strength	Strength1	Strength2
1	1	1	1	17	8	3	3
2	1	1	1	18	8	3	3
3	2	1	1	19	8	3	3
4	2	1	1	20	9	3	3
5	3	1	1	21	9	3	3
6	3	2	1	22	9	3	3
7	4	2	1	23	10	3	3

Table III.1/H.263 – TMN

QUANT	Strength	Strength1	Strength2	QUANT	Strength	Strength1	Strength2
8	4	2	2	24	10	4	3
9	4	2	2	25	10	4	3
10	5	2	2	26	11	4	3
11	5	3	2	27	11	4	3
12	6	3	2	28	11	4	3
13	6	3	2	29	12	4	3
14	7	3	2	30	12	4	3
15	7	3	3	31	12	4	3
16	7	3	3				

III.5.2 De-blocking post-filter

In this subclause, a two-dimensional regularized post filter to simultaneously reduce the blocking and ringing artefacts is described. This filter is recommended when Annex J is not used, so that over-blurred results can be avoided. The degradation model due to the quantization process can be written as:

$$g(i, j) = f(i, j) + n(i, j)$$

where *g, f, n* denote the reconstructed image, the original image, and quantization noise in spatial domain, and (i, j) represents the vertical and horizontal coordinates in the video picture. A four dimensional smoothness metric, *M*, is incorporated in order to minimize the quantization noise. *M* is defined as:

$$M(f(i, j)) = M_L(f(i, j)) + M_R(f(i, j)) + M_U(f(i, j)) + M_D(f(i, j))$$

$$M_L(f(i, j)) = (1 - \alpha_L(f(i, j)))[f(i, j) - g(i, j)]^2 + \alpha_L(f(i, j))[f(i, j) - f(i, j-1]^2$$

$$M_R(f(i, j)) = (1 - \alpha_R(f(i, j)))[f(i, j) - g(i, j)]^2 + \alpha_R(f(i, j))[f(i, j) - f(i, j+1)]^2$$

$$M_U(f(i, j)) = (1 - \alpha_U(f(i, j)))[f(i, j) - g(i, j)]^2 + \alpha_U(f(i, j))[f(i, j) - f(i, j-1, j)]^2$$

$$M_D(f(i, j)) = (1 - \alpha_D(f(i, j)))[f(i, j) - g(i, j)]^2 + \alpha_D(f(i, j))[f(i, j) - f(i, j+1, j)]^2$$

where α_i (i = L, R, U, D) represents the regularization parameters to control the relative contribution between fidelity and directional smoothness. Figure III.4 shows the example for the position of these pixels.

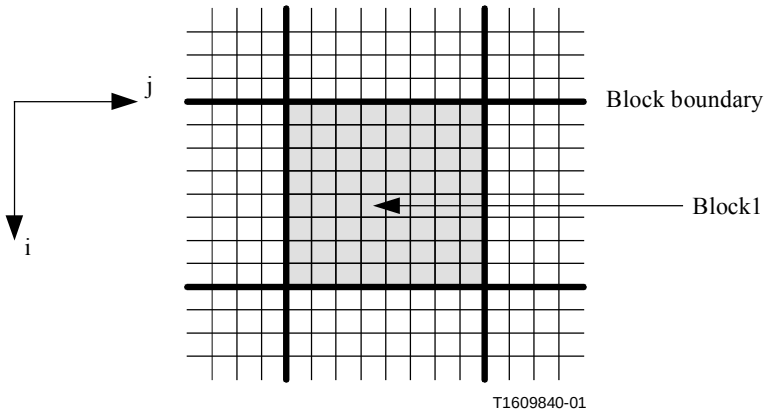


Figure III.4/H.263 – Example of pixels' positions

The COD bit is employed to estimate the correlation of a macroblock between a current frame and the previous coded picture. Therefore, different filters are used depending on COD, resulting in complexity reduction. The following describes the filters when COD is 0 (coded) or 1 (not coded).

Condition 1: block1 belongs to a coded macroblock (COD = 0)

If block1 shown in Figure III.4 (grey region) belongs to a coded macroblock, the following filter is applied to all pixels of block1. When pixel $g(i, j)$ belongs to block1, $f(i, j)$ is obtained by minimizing the following function:

$$f(i,j) = \frac{(4 - \alpha_{TOT}(f(i,j)))g(i,j) + A}{4}$$

$$\alpha_{TOT}(f(i,j)) = \alpha_L(f(i,j)) + \alpha_R(f(i,j)) + \alpha_U(f(i,j)) + \alpha_D(f(i,j))$$

$$A = \alpha_L(f(i,j))g(i,j-1) + \alpha_R(f(i,j))g(i,j+1) + \alpha_U(f(i,j))g(i-1,j) + \alpha_D(f(i,j))g(i+1,j)$$

The regularization parameters are chosen as:

$$\alpha_L(f(i,j)) = \frac{K_L QP_l^2}{[g(i,j) - g(i,j-1)]^2 + K_L QP_l^2}$$

$$\alpha_R(f(i,j)) = \frac{K_R QP_l^2}{[g(i,j) - g(i,j+1)]^2 + K_R QP_l^2}$$

$$\alpha_U(f(i,j)) = \frac{K_U QP_l^2}{[g(i,j) - g(i-1,j)]^2 + K_U QP_l^2}$$

$$\alpha_D(f(i,j)) = \frac{K_D QP_l^2}{[g(i,j) - g(i+1,j)]^2 + K_D QP_l^2}$$

where QP_l denotes the value of quantization parameter (quantization step size) of the last coded macroblock. Since larger smoothness is necessary at block boundaries and relatively smaller smoothness is required within a block, the constants, K_i (i = L, R, U, D) are determined depending on the position of $g(i, j)$ such as:

$$K_L, K_U = \begin{cases} 9 & \text{if } j \bmod 8 = 0 \\ 1 & \text{otherwise} \end{cases}$$

$$K_R, K_D = \begin{cases} 9 & \text{if } j \bmod 8 = 7 \\ 1 & \text{otherwise} \end{cases}$$

Condition 2: block1 belongs to an uncoded macroblock (COD = 1)

If block1 shown in Figure III.4 (shaded region) belongs to a uncoded macroblock, the following rule is applied to all pixels of block1.

$$f(i, j) = f_p(i, j)$$

where f_p represents the corresponding pixels in the recovered previous picture.

The following pseudo-code fragment can be used to ease the implementation of the described algorithm:

```
f   /* two dimensional array, recovered frame */
g   /* two dimensional array, reconstructed frame */
QP   /* quantization parameter */
K_L  /* left side constant for regularization parameter */
K_R  /* right side constant for regularization parameter */
K_U  /* up side constant for regularization parameter */
K_D  /* down side constant for regularization parameter */
alpha_L /* regularization parameter to left side */
alpha_R /* regularization parameter to right side */
alpha_U /* regularization parameter to up side */
alpha_D /* regularization parameter to down side */

QP=(QP*QP);
if(COD == 0)
{
     diff_L=(g(i,j)-g(i,j-1));
     diff_R=(g(i,j)-g(i,j+1));
     diff_U=(g(i,j)-g(i-1,j));
     diff_D=(g(i,j)-g(i+1,j));

     if(i mod 8 ==0){
          K_U=9;
          K_D=1;
     } else if(i mod 8 == 7){
          K_U=1;
          K_D=9;
     } else{
          K_U=1;
          K_D=1;
     }

     if(j mod 8 ==0){
          K_L=9;
          K_R=1;
     } else if (j mod 8 == 7){
          K_L=1;
          K_R=9;
     } else{
          K_L=1;
          K_R=1;
     }

     alpha_L=((K_L*QP)/(diff_L*diff_L+K_L*QP));
     alpha_R=((K_R*QP)/(diff_R*diff_R+K_R*QP));
     alpha_U=((K_U*QP)/(diff_U*diff_U+K_U*QP));
     alpha_D=((K_D*QP)/(diff_D*diff_D+K_D*QP));

     f(i,j)=(((4-alpha_L-alpha_R-alpha_U-alpha_D)*g(i,j)+alpha_L*g(i,j-
     )+alpha_R*g(i,j+1)+alpha_U*g(i-1,j)+alpha_D*g(i+1,j))/4);

}
```

III.5.3 Error detection

Parts of this clause are based in the TCON error concealment model described in LBC-95-186. In error-prone environments, it is necessary that the decoder be tolerant to erroneously received bitstreams. In practice, there are usually some mechanisms in the transport protocol hierarchy that provides information about damaged or lost parts of the bitstream, such as the RTP sequence number or the H.223 AL3 CRC check. By dividing coded pictures into (more or less) independently decodable segments, using GOB headers or slices, a reasonable performance can be achieved, even without basic mechanisms for error recovery in the video decoder itself.

For the error-prone application scenarios described in III.2.3 and III.2.4, bit errors are never present in the bitstream as the transport stack already assures bit error-free environments by the means of CRC checks. Therefore, the only possible syntactic error is missing information between synchronization markers (picture, GOB or slice headers). This error must be detected during decoding and error concealment as described in III.5.4 must be applied to the missing macroblocks. If a picture header is lost, it is replaced by the picture header of the previous picture, or by a redundant picture header if available (e.g. in the RTP payload header).

In the case where bit errors are present in the received bitstream, errors must be detected by the video decoder using syntax or semantic violations. These include:

- An illegal codeword is found. This is the most frequent event to stop decoding.
- A synchronization marker does not follow after reconstruction of a macroblock line. (If it is not known that GBSCs are used on every row of macroblocks, this condition cannot be tested.)
- Motion vectors point outside of the allowable range.
- Position of reconstructed DCT coefficient points are outside position 63, when Annex S is not in use.
- Chroma DC values are out of the normal range.

The first two bullets are the most important. The list of checkpoints could be increased considerably. When such an error is detected, the video decoder looks for the next synchronization point in the bitstream. All data between synchronization markers where the error was detected is discarded (i.e. not copied to the frame memory) and concealed as described in III.5.4.

III.5.4 Error concealment

It is believed that the following algorithm is efficient whenever Annex J is in use and the appropriate postfilter, as defined in III.5.1. Interoperability with the postfilter discussed in III.5.2 is under study.

Parts of this clause are based in the TCON error concealment model described in LBC-95-186. Missing macroblocks are concealed using a very simple, but effective algorithm. In parts of the picture where data is lost, data from the previously decoded picture is used for concealment. No error concealment is applied if part of the first received picture is corrupted. The motion vector of the missing macroblock is set to the motion vector of the macroblock above the missing macroblock. If this motion vector is not available, the motion vector is set to *zero*. Using the appropriate motion vector, the macroblock from the previous picture at the spatial location specified by this motion vector is copied to the location of the missing macroblock in the current picture.

Special cases for 4MV and OBMC:

- If the macroblock above the missing macroblock was coded with four motion vectors (as could be the case with use of Annexes F or J), the error concealment operates in the same way on an 8×8 block basis, so that:
  - the two 8×8 blocks on the left side of the missing macroblock use the motion vector of the 8×8 block on the left side of the bottom of the macroblock above; and similarly

– the two 8×8 blocks on the right side of the missing macroblock use the motion vector of the 8×8 block on the right side of the bottom of the macroblock above.

- If Annex F is in use, the OBMC weighting is applied using the concealment motion vectors.

III.6 Information capture section

This clause contains all information that was officially approved for adoption into the test model by the ITU-T video coding experts group, but which does not fit into the application scenarios defined in III.2. Most of this information applies only to situations when certain optional modes are used, such as the scalability mode (Annex O), PB-frames (Annexes G and M), or the reduced resolution update mode (Annex Q). Not all of the information here is thoroughly checked to be compatible with the 'mainstream' algorithms used for the application scenarios. Additional information is solicited.

III.6.1 PB-frames mode (Annexes G and M)

Annex M provides better prediction options than Annex G, and should always be employed instead of Annex G whenever H.263 Version 2 is employed.

III.6.1.1 Improved PB-frames motion estimation and mode decision

The candidate forward and backward motion vectors for each of the blocks in the B-macroblock is obtained by scaling the best motion vector from the P-macroblock, MV, as specified in H.263. To find the SADbidir, these vectors are used to perform a bidirectional prediction, as described in Annex M, improved PB-frames, in the H.263 standard, but with MVD set to zero.

Then, for the 16×16 B-macroblock, a normal integer and half-pixel motion estimation is performed, relative to the previous reconstructed P-picture. The best SADforw for this motion estimation is compared with the SADbidir for the bidirectional prediction. If (SADforw < SADbidir – 100), forward prediction is chosen for this macroblock. In this case, the forward motion vector found in the motion estimation above, is transmitted directly in MVDB, with no motion vector prediction. If the bidirectional prediction is found to be the best, no MVDB is transmitted.

III.6.2 Scalability mode (Annex O)

III.6.2.1 True B-frame motion estimation and mode decision

B-frames have several permissible prediction options, therefore coding modes. Forward motion estimation is performed relative to the previous reconstructed I/P-frame, and backward motion estimation is performed relative to the future reconstructed I/P-frame.

In addition, B-frames allow for direct mode prediction, which does not require motion vector data to be transmitted, and bidirectional mode prediction, which uses the resulting motion vectors of the forward and backward motion estimation. For direct mode prediction, the motion vector between the surrounding I/P- or P/P-frame pair is scaled.

The preferred coding modes, in order, are direct, forward, backward and bidirectional. To reflect the preferences, the SAD is computed for each possible prediction mode then adjusted. The direct mode is favoured by subtracting 100 from its SAD. The forward mode is favoured by subtracting 50 from its SAD. The backward mode SAD is unchanged. The bidirectional mode SAD is penalized by adding 75. The mode with the lowest SAD after these modifications is chosen as the coding mode.

III.6.2.2 EI/EP-frame motion estimation and mode decision

EI- and EP-frames also have several permissible prediction options, therefore coding modes. Forward motion estimation is performed relative to the previous reconstructed EI/EP-frame.

In addition, EI- and EP-frames allow for upward mode prediction, which does not require motion vector data to be transmitted, and bidirectional mode prediction, which uses the resulting motion

vectors of the forward motion estimation. For upward mode prediction, the prediction is made from the same spatial location of the reconstructed, and possibly upsampled, frame the reference layer.

The preferred coding modes, in order, are upward, forward and bidirectional. To reflect the preferences, the SAD is computed for each possible prediction mode then adjusted. The upward mode is favoured by subtracting 50 from its SAD. The forward mode SAD is unchanged. The bidirectional mode SAD is penalized by adding 100. The mode with the lowest SAD after these modifications is chosen as the coding mode.

III.6.2.3 Rate control for P- and B-frames

III.6.2.3.1 Macroblock level

The macroblock level rate control described in III.4.2.2 can be used directly for B-frames. The only difference is that since the statistics of B-frames are different from those of P-frames, the rate control parameters K and C (which are updated at each macroblock) take values in different ranges. Consequently, when using P- and B-frames, different parameters $\{K_P, C_P\}$ and $\{K_B, C_B\}$ for the P- and B-frames, respectively, are to be used.

III.6.2.3.2 Frame level

The frame-level rate control in III.4.2.1 assigns a near constant target number of bits per P-frame (after the first I frame), which is an effective strategy for low-delay video communications. But in scenarios where one or several B-frames are inserted between the Ps, since the B-frames are easier to encode, some technique is needed to assign fewer bits to the B-frames.

In this subclause, an appropriate technique for assigning target number of bits to P- and B-frames is described. The derivation of this method is discussed in [Q15-C-19] [Q15-D-22]. The derivation is based on the typical case where the pattern of frame types is:

I, B, …, B, P, B, …, B, P, B, …, B, P, B, …, B, P, …

The set of frames "B, …, B, P" is repeated periodically after the first I frame and is referred to as a group of pictures or GOP. Let M_B be the number of B-frames in a GOP. The target number of bits for the P picture in that GOP, T_P, and the target for each of the B-frames, T_B, can be computed as follows:

$$T_P = T - M_B T_B$$

$$T_B = \frac{T - 16^2 N(C_P - \beta\, C_B)}{\beta + M_B}$$

$$\beta = 0.9\, \beta_{PREV} + 0.1F \frac{E_P}{E_C}$$

where the parameters in the above equations are defined as:

- T, M, and N are, respectively, the number of bits for the GOP, the number of frames for the GOP, and the number of macroblocks in a frame.
- The value of β determines how many bits are assigned to the P-frame and how many are assigned to the Bs. β increases with F and E_P/E_B, which is described below.
- F determines how large the PSNR of the P-frames is in comparison to that of the Bs. For example, if F is equal to 1, the PSNR of both types of frames will be similar and if F is larger than 1 the PSNR of the Ps increases with respect to that of the Bs. The following formula is used to determine the value of F:

$$F = \max\left\{\min\left(\frac{1.4}{\sqrt{\text{Bpp}}} - 0.3,\ 5\right),\ 1\right\}$$

where Bpp is the rate in bits per pixel for the video sequence. Using this relationship, the PSNR of the P-frames is on average about 1 dB higher than that of the B-frames, which appears to be a reasonable trade-off.

- E_P is the energy for the P-frame in the previous GOP, where energy is defined as the sum of the variances of the macroblock prediction errors, i.e.

$$E_P = \sum_{i=1}^{N} \sigma_i^2$$

where σ_i^2 is the variance of the i-th macroblock in the (previous) P-frame, as defined in III.4.2.2. On the other hand, E_B is the mean of the energies for the B-frames in the previous GOP, i.e.

$$E_B = \frac{1}{M_B} \sum_{m=1}^{M_B} E_{B,m}$$

where $E_{B,m}$ is the energy of the mth B-frame in the previous GOP.

- β_{prev} is set to F for the first GOP and to the previous value of β for the next GOPs.
- C_P and C_B are the motion and syntax rate (in bits per pixel) for the P- and B-frames, respectively, and their values are obtained from the rate control at the respective macroblock levels as described in III.4.2.2.

The previous frame-level rate control described in III.4.2.1, was designed for GOPs of the type "P…P", and corresponds to the special case where $E_P = E_B$, $F = 1$, (or, equivalently, $\beta = 1$) and $C_P = C_B$ in the above equations.

Finally, before a given frame is encoded (with a target of either T_P or T_B bits), the value Δ is subtracted from the target bit rate, as defined in III.4.2.1, which provides feedback from the fullness of the encoder buffer and the frame skipping threshold. The latter was set to the channel bit rate (in bits per second) divided by the encoding frame rate, which is a good choice for low-delay scenarios. But when B frames are inserted between the Ps, and hence delay is not as important, a larger frame skipping threshold (and larger value of A in III.4.2.1) would be more appropriate.

III.6.2.3.3 SNR and spatial enhancement layer rate control

Typically the bit rate available for each of the enhancement layers is determined by the specific application. At each layer, the rate control can be used equivalently as in the base layer with the bit rate, frame rate, and GOP pattern for the given layer. The only difference is that, since different frames have different statistics at different layers, there must be different variables for K and C at each layer. Specifically, there should be different parameters K and C for different layers and, within a layer, different K and Cs for each frame type.

III.6.2.4 Usage in error-prone environments

In addition to other error resilience methods introduced in this appendix, Picture Numbers defined in Annex U (Enhanced Reference Picture Selection) and in Annex W (Additional Supplemental Enhancement Information Specification) provide an error control method especially suitable for Annex O. Picture Numbers are an integral part of the bitstream when Annex U is in use. Otherwise, it is recommended to use Picture Numbers as defined in Annex W.

The definition of Picture Numbers in Annexes U and W can be concluded as follows: Picture Number shall be incremented by one for each coded and transmitted I, P, PB, EI, and EP picture, relative to the previous stored picture in the same enhancement layer. For B pictures, Picture Number shall be incremented relative to the value in the most recent stored non-B picture in the reference layer of the B picture which precedes the B picture in bitstream order (a picture which is temporally subsequent to the B picture). If adjacent pictures in the same enhancement layer have the same temporal reference, and if a reference picture selection mode is in use, the decoder shall regard this occurrence as an indication that redundant copies have been sent of approximately the same pictured scene content, and all of these pictures shall share the same Picture Number.

If the difference of the Picture Numbers of two consecutively received and stored pictures in the same enhancement layer is not one, and if the pictures do not represent approximately the same pictured scene content as described above, the decoder should infer a loss of pictures or corruption of data. In such a case, a back-channel message as defined in Annex U or other similar indication signalling the loss of pictures may be sent to the far-end encoder. If the lost pictures resided in one of the enhancement layers, the decoder can go on decoding the base layer and the enhancement layers lower than the corrupted one. The decoder may also carry on decoding the layer that suffered from the loss of pictures and any layer above that especially if a suitable spare reference picture (see W.6.3.13/H.263) is signalled.

III.6.3 Reduced-resolution update mode (Annex Q)

This clause describes how to use the reduced-resolution update mode, defined in Annex Q. The information here has not been verified.

III.6.3.1 Motion estimation and mode selection

In reduced-resolution update mode, motion estimation is performed on the luminance 32×32 macroblock instead of 16×16 macroblock. SAD (Sum of Absolute Difference) is used as an error measure. In this mode, each component of the macroblock motion vector or four motion vectors is restricted to be half-integer or zero value in order to widen the search range with the same MVD table.

III.6.3.2 Motion estimation in baseline mode (no options)

III.6.3.2.1 Integer-pixel motion estimation

The search is made with integer-pixel displacement in the Y component. The comparisons are made between the incoming macroblock and the displaced macroblock in the previous reconstructed picture. A full search is used, and the search area is up to **±30** pixels in the horizontal and vertical direction around the original macro block position.

$$SAD(x,y) = \sum_{i=1,\,j=1}^{32,32} |original - decoded\_previous|, x, y = \text{"up to } \pm 30\text{"}$$

For the zero vector, SAD(0,0) is reduced by **400** to favour the zero vector when there is no significant difference.

$$\text{SAD}\ (0,0) = \text{SAD}\ (0,0) - 400$$

The (x, y) pair resulting in the lowest SAD is chosen as the integer-pixel motion vector, MV0. The corresponding SAD is SAD(x,y).

III.6.3.2.2 INTRA/INTER mode decision

After the integer-pixel motion estimation the coder makes a decision on whether to use INTRA or INTER prediction in the coding. The following parameters are calculated to make the INTRA/INTER decision:

$$MB\_mean = \frac{\sum_{i=1,j=1}^{32,32} original}{1024}$$

$$A = \sum_{i=1,j=1}^{32,32} |original - MB\_mean|$$

INTRA mode is chosen if: $A < (SAD(x,y) - 2000)$.

Notice that if SAD(0,0) is used, this is the value that is already reduced by 400 above.

If INTRA mode is chosen, no further operations are necessary for the motion search. If INTER mode is chosen the motion search continues with half-pixel search around the MV0 position.

III.6.3.2.3 Half-pixel search

The half-pixel search is done using the previous reconstructed frame. The search is performed on the Y-component of the macroblock. The search area is ±1 half-pixel around the 32×32 target matrix pointed to by MV0, complying with the condition that each component of the candidate vector for the half-pixel search is half-integer or zero value. For the zero vector (0,0), SAD(0,0) is reduced by 400 as for the integer search.

The half-pixel values are calculated as described in 6.1.2/H.263.

The vector resulting in the best match during the half-pixel search is named MV. MV consists of horizontal and vertical components (MVx, MVy), both measured in half-pixel units.

III.6.3.3 Motion estimation in advanced prediction mode (Annex F)

This clause applies only if advanced prediction mode is selected.

III.6.3.3.1 Integer-pixel motion estimation

±2 integer-pixel search within [−31, 30] is performed for the **16 × 16 blocks** around 32×32 integer vector.

III.6.3.3.2 Half-pixel search

The half-pixel search is performed for each of the blocks around the 16×16 integer vector. The search area is ±0.5 pixel around the 16×16 integer vector of the corresponding block, complying the condition that each component of the candidate vector for the half-pixel search is half-integer or zero value and within [−31.5, 30.5].

III.6.3.3.3 One vs four MV decision in AP

This subclause applies only if advanced prediction mode is selected.

SAD for the best half-pixel **32 × 32 MB** vector (including subtraction of **400** if the vector is (0,0)):

$$SAD_{32}(x,y)$$

SAD for the whole macroblock for the best half-pixel **16 × 16 block** vectors:

$$SAD_{4\times16} = \sum_{1}^{4} SAD_{16}(x, y)$$

The following rule applies:

If: $SAD_{4\times16} < SAD_{32} - 800$, choose **16 × 16 block** prediction

otherwise: choose **32 × 32 MB** prediction.

III.6.3.4 Motion estimation in the unrestricted motion vector mode (Annex D)

This clause applies only if the extended motion vector range in the UMV mode is selected.

III.6.3.4.1 Search window limitation

Since the window with legal motion vectors in this mode is centred around the motion vector predictor for the current macroblock, some restrictions on the integer motion vector search is applied, to make sure the motion vectors found will be transmittable.

With these restrictions, both the **32 × 32 MB** vector and the **16 × 16 block** vectors found with the procedure described below, will be transmittable, no matter what the actual half-pixel accuracy motion vector predictor for the macroblock, or each of the four blocks, turns out to be.

III.6.3.4.2 Integer-pixel search

First, the motion vector predictor for the **32 × 32 MB** vector based on integer motion vectors only, is found. The **32 × 32 MB** search is then centred around the truncated predictor, with a somewhat limited search window. If four vectors, the **32 × 32 MB** search window is limited to the range **29 – (2*16 × 16_block_search_window + 1)**. Since in this model the **16 × 16_block** search window is **2.5**, the default search window of **32 × 32 MB** in the UMV mode turns out to be **±23** integer positions. Then the **16 × 16_block** searches are centred around the best **32 × 32 MB** vector, and **±2** pixel search is performed in each **16 × 16_block**.

III.6.3.4.3 Half-pixel search

Half-pixel searches are performed as in the other modes. The search area is ±0.5 pixel around the best integer vector of the corresponding macroblock/block, complying the condition that each component of the candidate vector for the half-pixel search is half-integer or zero value.

III.6.3.5 Down-sampling of the prediction error

After motion compensation on 16 × 16 block basis, the 16 × 16 prediction error block is down-sampled to the 8 × 8 reduced-resolution prediction error block. In order to realize a simple implementation, filtering is constrained to a block which enables up-sampling on an individual block basis. Figure III.5 shows the positioning of samples. The down-sampling procedure for the luminance and chrominance-pixels is defined in Figure III.6. Filtering is performed regardless of the block boundary. "/" in Figure III.6 indicates division by truncation.

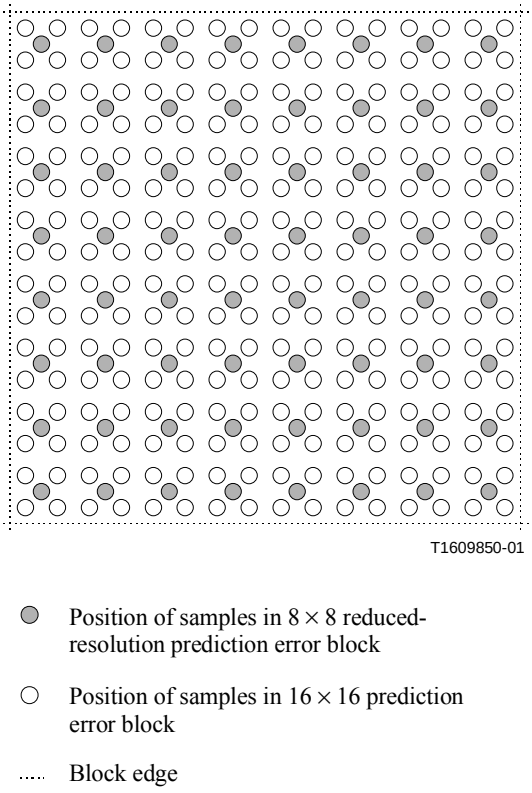


Figure III.5/H.263 – Positioning of samples in 8 × 8 reduced-resolution prediction error block and 16 × 16 prediction error block

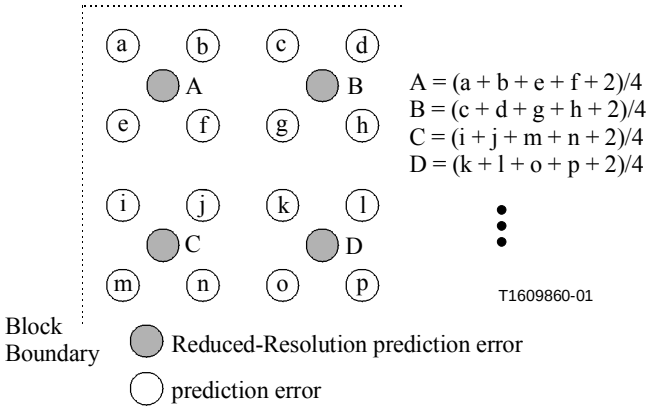


Figure III.6/H.263 – Creation of reduced-resolution prediction error for pixels inside block

III.6.3.6 Transform and quantization

A separable two-dimensional Discrete Cosine Transform (DCT) is applied to the 8 × 8 reduced-resolution prediction error block in the same way as the default mode. Then quantization is performed in the same way as described in the default mode.

III.6.3.7 Switching

In this clause, a simple switching algorithm for Annex Q is described.

NOTE – This algorithm might be applicable to "Factor of 4" part of Annex P with a small modification.

III.6.3.7.1 Resolution decision algorithm

In order to decide the resolution for Annex Q, a simple decision algorithm is used based on $\overline{QP}_{i-1}$ and B_{i-1}.

$\overline{QP}_{i-1}$ the mean QP of the previous encoded frame;

B_{i-1} the number of bits used in the previous encoded frame.

Assuming that the relation between $\overline{QP}_{i-1}$ and B_{i-1} is close to inverse proportion, the product of $\overline{QP}_{i-1}$ and B_{i-1} can be regarded as an index of the approximate complexity of the coded frame.

In the case the current frame is default mode, switching to reduced-resolution update mode is done if the product of $\overline{QP}_{i-1}$ and B_{i-1} is larger than the threshold TH1.

In the case the current frame is reduced-resolution update, switching to default mode is done if this product of $\overline{QP}_{i-1}$ and B_{i-1} is smaller than the threshold TH2.

From **default mode** to **reduced-resolution update mode**

if($\overline{QP}_{i-1} * B_{i-1} > TH1$){
    Switch to
    reduced-resolution update mode;
}

From **reduced-resolution update mode** to **default mode**

if($\overline{QP}_{i-1} * B_{i-1} < TH2$){
    Switch to ***default mode;***
}

TH1 is determined in the following equation, where QP1 and FR1 represents the lowest subjective quality which is permitted to be encoded in default mode.

TH1 = QP1 * (Target_Bit rate / FR1)

In the same way, TH2 is determined in the following equation, where QP2 and FR2 represents the highest subjective quality which is permitted to be encoded in reduced-resolution update mode.

TH2 = QP2 * (Target_Bit rate / FR2)

The values of QP1, FR1, QP2, and FR2 may depend on the source format, target frame rate, and target bit rate. If Source Format indicates CIF, the target frame rate is 10 fps, and the target bit rate is 48 kbit/s, then QP1 = 16, FR1 = 7, QP2 = 7, FR2 = 9, respectively.

III.6.3.8 Restriction of DCT coefficients in switching from reduced-resolution to normal-resolution

Once the reduced-resolution mode is selected, the detail of the image is likely to be lost. If the mode goes back to the default mode again, the detail of the image must be reproduced, which consumes a large amount of bits. This sudden increase of coding bits often causes an unintentional frame skips. Furthermore, because the resolution-decision algorithm described above uses the product of mean QP and the amounts of bits, this sudden increase of the bits cause to switch back to reduced-resolution update mode, and the oscillation between both modes often occurs. In order to avoid this degradation, the restriction of DCT coefficients to be sent is introduced to the several frames after switching from reduced-resolution update mode to default mode. In the first frame after switching to the default mode, the coefficients only within 4×4 low frequency can be sent, then in

the same way, 5×5 in the second, 6×6 in the third, and 7×7 in the forth. This "smooth-landing" algorithm can suppress the unintentional frame skip and the oscillation of the modes effectively.

III.6.3.8.1 Rate control

The rate control is identical to the default mode, except the quarter number of Macroblocks.

III.6.4 Fast search using mathematical inequalities

Several methods are known to speed-up motion search that are based on mathematical inequalities [LS95] and [LT97]. These inequalities, e.g. the triangle inequality, give a lower bound on the norm of the difference between vectors. In block matching, the search criteria very often used for the distortion are the sum of the absolute differences (SAD) or the sum of the squared differences (SSD) between the motion-compensated prediction c[x, y] and the original signal s[x, y]. Incorporating the triangle inequality into the sums for SAD and SSD yields:

$$D(s,c) = \sum_{[x,y] \in B} \left| s[x,y] - c[x,y] \right|^p \geq$$

$$\hat{D}(s,c) = \left| \left(\sum_{[x,y] \in B} \left| s[x,y] \right|^p \right)^{1/p} - \left(\sum_{[x,y] \in B} \left| c[x,y] \right|^p \right)^{1/p} \right|^p \qquad \text{(III-6.4.1)}$$

by varying the parameter $p = 1$ for SAD and $p = 2$ for SSD. Note that for $p = 2$, the inequality used in [LT97] differs from Equation (III-6.4.1). Empirically, little difference has been found between those inequalities. For some blocks, the inequality used in [LT97] provides a more accurate bound whereas for other blocks the triangle inequality performs better. The set B comprises the sampling positions of the blocks considered, e.g. a block of 16×16 samples.

Assume D_{min} to be the smallest distortion value previously computed in the block motion search. Then, the distortion $D(s,c)$ of another block c in our search range is guaranteed to exceed D_{min} if the lower bound of $D(s,c)$ exceeds D_{min}. More precisely, reject block c if:

$$\hat{D}(s,c) \geq \mathrm{D}_{\min} \qquad \text{(III-6.4.2)}$$

The special structure of the motion estimation problem permits a fast method to compute the norm values of all blocks $c[x, y]$ in the previously decoded frames [LS95]. The extension to a rate-constrained motion estimation criterion is straightforward [CM97].

III.6.4.1 Search order

It is obvious that a small value for D_{min} determined in the beginning of the search leads to the rejection of many other blocks later and thus reduces computation. Hence, the order in which the blocks in the search range are checked has a high impact on the computation time. For example, given the Huffman code tables for the motion vectors as prior information about our search space, the search ordering should follow increasing bit rate for the motion vectors. This increases the probability of finding a good match early in the search process. A good approximation of these probabilities is the spiral search, described in III.3.1.3.1.

III.6.4.2 Multiple triangle inequalities

Following [WZG99], multiple triangle inequalities can be employed. Assume a partition of B into subsets B_n so that:

$$B = \bigcup_n B_n \text{ and } \bigcap_n B_n = \phi \qquad \text{(III-6.4.3)}$$

The triangle inequality (III-6.4.1) holds for all possible subsets B_n. Rewriting the formula for $D(s, c)$ yields:

$$\sum_{[x,y]\in B}|s[x,y]-c[x,y]|^p=\sum_n\sum_{[x,y]\in B_n}|s[x,y]-c[x,y]|^p \qquad \text{(III-6.4.4)}$$

and applying the triangle inequality for all B_n yields:

$$D(s,c)=\sum_{[x,y]\in B}|s[x,y]-c[x,y]|^p\geq\sum_n\left|\left(\sum_{[x,y]\in B}|s[x,y]|^p\right)^{1/p}-\left(\sum_{[x,y]\in B}|c[x,y]|^{1/p}\right)\right|^p \qquad \text{(III-6.4.5)}$$

Note that equation (III-6.4.5) is a tighter lower bound than equation (III-6.4.1); however, it requires more computation. Hence, this allows for the trade-off of sharpness of the lower bound against computational complexity.

An important issue within this context remains to be the choice of the partitions B_n. Of course, (III-6.4.5) works for all possible subsets that satisfy (III-6.4.4). However, since the norm values of all blocks in the search space have to be pre-computed, the fast method described in [LS95] can be applied. Therefore, a random sub-division of B into n arbitrary subsets may not be the appropriate choice. Instead, for the sake of computation, a symmetric sub-division of B may be more desirable. In [LT97], it is proposed to divide a square 16×16 block into two different partitions. The first partitioning produces 16 subsets B_n each being one of 16 lines containing 16 samples. The second partition consists of 16 subsets B_n each being one of 16 columns containing 16 samples.

Since H.263 permits 8×8 blocks, the approach proposed in [LC95] is applied, where a 16×16 block is decomposed into sub-blocks. The 16×16 block is partitioned into one set of 16×16 samples, into four subsets of 8×8 samples. The various (subset) triangle inequalities are successively applied in the order of the computation time to evaluate them, i.e. first the 16×16 triangle inequality is checked, then the inequalities relating to blocks of size 8×8. On the 8×8 block level, the 8×8 triangle inequality is checked only.

III.6.5 Control of encoding frame rate

This clause contains two algorithms that can be used instead of the frame layer rate control algorithm defined in III.4.2.1. While the fixed frame rate algorithm is useful only in environments where a completely fixed frame rate is more important than perceptual quality and delay, the adaptive control algorithm increases the perceptual quality while maintaining reasonable delay characteristics by adjusting the frame rate according to the scenes content.

III.6.5.1 Adaptive control of the encoding frame rate

The objective of this control of encoding frame rate algorithm is to keep the quality of P-frames in the tolerable range under sudden motion change and time-varying communication channel environments without obvious degradation in the perceptual motion smoothness. The algorithm adjusts the encoding frame rate adaptively based on the motion information in the underlying video to keep the image quality of each P-frame in a tolerable range. Since it is difficult to support good quality in both spatial and temporal resolution (in terms of motion smoothness) at very low bit rates, a control of encoding frame rate is adopted for a trade-off of spatial/temporal quality based on the motion in video and the available channel bandwidth. It is observed that human eyes are sensitive to the abrupt encoding frame rate (or interval) change. This scheme aims at the reduction of temporal degradation in terms of motion jerkiness perceived by human beings. At the same time, low encoding time-delay is imposed for real-time processing. The control of encoding frame rate under time-varying CBR and the relation with so called TMN8 rate control algorithm of III.4.2 is explained in the following. The development of this algorithm is described in [Q15-G-22].

The following definitions are used.

i: Encoded frame index.

a, b, a' and b': Frame layer R-D model coefficients.

f_{i-1}: Reconstructed reference frame at the previous time instant.

f_i: Uncompressed image at the current time instant.

$\overline{QP}_i$: Average QP of all macroblocks in a frame.

$\hat{B}(Q), \hat{D}(Q)$: Rate and distortion models of a frame respectively.

$MAD(f_{i-1}, f_i)$: Mean of absolute difference between f_{i-1} and f_i.

R_i: Current available channel bandwidth.

F_i: Current encoding frame interval under the assumption the camera captures frames at a rate of G fps.

Step 0: Set the initial parameters such as initial encoding frame rate F_0, encoding frame interval adjustment threshold c = 0.04. Also, frame layer R-D table size T_0, T_{max} are set to 0 and 20, respectively. Finally the iteration starts with $i = 1$.

Step 1: Shift the R-D database table by setting $B_k = B_{k+1}, D_k = D_{k+1}, \overline{QP}_k = \overline{QP}_{k+1}$ for $k = 1, ..., T_{i-1} - 1$.

Then, add a new item to the end of table by $B_{T_{i-1}} = B_{i-1}, D_{T_{i-1}} = D_{i-1}, \overline{QP}_{T_{i-1}} = \overline{QP_{i-1}}$.

Step 2: Derive rate and distortion models with respect to the average QP of frames. First, calculate the rate and distortion model coefficients by using the above R-D table.

$$\begin{aligned} \hat{B}(QP_i) &= \left(a \cdot QP_i^{-1} + b \cdot QP_i^{-2}\right) \cdot MAD(f_{i-1}, f_i), \\ \hat{D}(QP_i) &= a'QP_i + b'. \end{aligned} \tag{III-6.5.1}$$

$$a = \frac{\sum_{k=1}^{T_{i-1}} \left\{R_k QP_k^{-1} - bQP_k^{-3}\right\}}{\sum_{k=1}^{T_{i-1}} QP_k^{-2}}, b = \frac{\left(\sum_{k=1}^{T_{i-1}} R_k QP_k^{-2}\right)\left(\sum_{k=1}^{T_{i-1}} QP_k^{-2}\right) - \left(\sum_{k=1}^{T_{i-1}} R_k QP_k^{-1}\right)\left(\sum_{k=1}^{T_{i-1}} QP_k^{-3}\right)}{\left(\sum_{k=1}^{T_{i-1}} QP_k^{-4}\right)\left(\sum_{k=1}^{T_{i-1}} QP_k^{-2}\right) - \left(\sum_{k=1}^{T_{i-1}} QP_k^{-3}\right)}, \tag{III-6.5.2}$$

$$a' = \frac{\sum_{k=1}^{T_{i-1}} d_k \sum_{k=1}^{T_{i-1}} QP_k - T_{i-1} \sum_{k=1}^{T_{i-1}} d_k QP_k}{\left(\sum_{k=1}^{T_{i-1}} QP_k\right)^2 - T_{i-1} \sum_{k=1}^{T_{i-1}} QP_k^2}, b' = \frac{\sum_{k=1}^{T_{i-1}} d_k - a' \sum_{k=1}^{T_{i-1}} QP_k}{T_{i-1}}, \tag{III-6.5.3}$$

Then, remove outlier utilizing the following check for $k = 1 : T_{i-1}$. If $\left|\hat{B} - B_k\right| > \sigma_B$ or $\left|\hat{D} - D_k\right| > \sigma_D$ then disable this datum temporarily and repeat Step 2 again with the refined data.

Step 3: Calculate the estimated distortion by:

$$\hat{D} = a' \frac{a \cdot MAD(f_{i-1}, f_i) + \sqrt{(a \cdot MAD(f_{i-1}, f_i))^2 + 4b \cdot B(F_{i-1}) \cdot MAD(f_{i-1}, f_i)}}{2B(F_{i-1})} + b', \quad \text{(III-6.5.4)}$$

$$B(F_{i-1}) = \frac{F_{i-1}}{G} \cdot R_i.$$

Step 4: To determine the encoding frame interval, first calculate the threshold values by $TH_{d1} = (1+c) \cdot D_{avg}$, $TH_{d2} = (1-c/2) \cdot D_{avg}$, where D_{avg} is the average distortion of the previous five encoded frames. Note that, for large *c*, control of encoding frame rate is turned off and only macroblock layer rate control of III.4.2.2 works. Then, adjust the encoding frame interval as shown in equation (III-6.5.5) with $\Delta F_{i-1} = \lfloor 0.3 \cdot F_{i-1} \rfloor$.

Step 5: Call macroblock level rate control of III.4.2.2 with the target bit rate $\tilde{B}_i$, which will return $\overline{QP}_i, B_i$ and D_i.

$$\tilde{B}_i = \begin{cases} (F_{i-1} + \Delta F_{i-1}) \cdot R_i / G & \text{if } \hat{D} > TH_{d1}, \\ (F_{i-1} + \Delta F_{i-1}) \cdot R_i / G & \text{if } \hat{D} < TH_{d2}, \\ F_{i-1} \cdot R_i / G & \text{otherwise} \end{cases} \quad \text{(III-6.5.5)}$$

Step 6: If all frames are encoded, then stop. Otherwise go to Step 1 with $T_i = \min\{T_{i-1} + 1, T_{\max}\}$ and $i = i + 1$.

III.6.6 Remarks on optimized use of the enhanced reference picture selection mode

III.6.6.1 Fast search techniques

Since motion estimation is extended from one to several pictures, fast search techniques are recommended. Please refer to document [Q15-D-55] for details on possible reductions in computation time.

III.6.6.2 Error resilient encoding

As for single-picture MCP, losses of picture content and concealment may result in different reference pictures at encoder and decoder for multi-picture MCP causing temporal error propagation. In both schemes, coding macroblocks in INTRA mode can stop temporal error propagation. But in multi-picture MCP, the choice of the motion vector and picture reference parameter can also have a significant impact on performance. For details on a possible encoding strategy please refer to document [Q15-H-24].

III.7 References

[CM97] COBAN (M.), MERSEREAU (R.M.): Computationally Efficient Exhaustive Search Algorithm for Rate-Constrained Motion Estimation, in *Proc. ICIP*, Santa Barbara, USA, Oct. 1997.

[GCK99] GALLANT (M.), CÔTÉ (G.), KOSSENTINI (F.): A Computation Constrained Block-based Motion Estimation Algorithm for Low Bit-rate Video Coding, *IEEE Transactions on Image Processing*, Dec. 1999.

[GFS97] GIROD (B.), FÄRBER (N.), STEINBACH (E.): Performance of the H.263 Video Compression Standard, *Journal of VLSI Signal Processing: Systems for Signal, Image, and Video Technology. Special Issue on Recent Development in Video:*

Algorithms, Implementation and Applications. Vol. 17, No. 2/3, pp. 101-111, Nov. 1997.

[LBC-95-186] Telenor Research: Definition of an Error Concealment Model (TCON), *Contribution LBC-95-186 to meeting of ITU-T SGXV Experts Group for Very Low Bit Rate Visual Telephony (Study Period 1993-1996),* Boston, June 1995.

[LC95] LEE (C.-H.), CHEN (L.-H.): A Fast Search Algorithm for Vector Quantization Using Mean Pyramids of Codewords, in *IEEE TR-COM*, Vol. 43, No. 2/3/4, pp. 604-612, Feb./March/April 1995.

[LS95] LI (W.), SALARI (E.): Successive Elimination Algorithm for Motion Estimation, *IEEE Trans. Image Proc.*, pp.105-107, Jan. 1995.

[LT97] LIN (Y.-C.), TAI (S.-C.): Fast Full-Search Block-Matching Algorithm for Motion-compensated Video Compression, in *IEEE TR-COM*, Vol. 45, No. 5, pp. 527-531, May 1997.

[Q15-A-20] RIBAS-CORBERA (J.), LEI (S.): Rate Control for Low-Delay Video Communications, *Contribution Q15-A-20 to ITU-T Video Coding Experts Group First Meeting (ITU-T SG16 Q.15 Study Period 1997-2000)*, Portland, Oregon, USA, June 1997.

[Q15-B-23] CÔTÉ (G.), GALLANT (M.), KOSSENTINI (F.): Experimental Results for Integer Pixel Fast Search Motion Estimation of TMN 8, *Contribution Q15-B-23 to ITU-T Video Coding Experts Group Second Meeting (ITU-T SG16 Q.15 Study Period 1997-2000)*, Sun River, Oregon, USA, Sept. 1997.

[Q15-C-19] RIBAS-CORBERA (J.), LEI (S.): Extension of TMN8 rate control to B frames and enhancement layer, *Contribution Q15-C-19 to ITU-T Video Coding Experts Group Third Meeting (ITU-T SG16 Q.15 Study Period 1997-2000)*, Eibsee, Germany, Dec. 1997.

[Q15-D-13] WIEGAND (T.), ANDREWS (B.): An Improved H.263 Coder Using Rate-Distortion Optimization, *Contribution Q15-D-13 to ITU-T Video Coding Experts Group Fourth Meeting (ITU-T SG16 Q.15 Study Period 1997-2000)*, Tampere, Finland, April 1998.

[Q15-D-22] RIBAS-CORBERA (J.), LEI (S.): An improvement on the extension of TMN8 rate control to B frames, *Contribution Q15-D-22 to ITU-T Video Coding Experts Group Fourth Meeting (ITU-T SG16 Q.15 Study Period 1997-2000)*, Tampere, Finland, April 1998.

[Q15-D-55] WIEGAND (T.), GIROD (B.), LINCOLN (B.), ANDREWS (B.): Fast Search for Long-Term Memory Motion-Compensated Prediction, *Contribution Q15-D-55 to ITU-T Video Coding Experts Group Fourth Meeting (ITU-T SG16 Q.15 Study Period 1997-2000)*, Tampere, Finland, April 1998.

[Q15-D-58] CÔTÉ (G.), EROL (B.), GALLANT (M.), KOSSENTINI (F.): H.263+: Video Coding at Low Bit Rates, *Contribution Q15-D-58 to ITU-T Video Coding Experts Group Fourth Meeting (ITU-T SG16 Q.15 Study Period 1997-2000)*, Tampere, Finland, April 1998.

[Q15-E-15] WENGER (S.), CÔTÉ (G.): Intra-Macroblock Refresh in Packet (Picture) Lossy Scenarios, *Contribution Q15-E-15 to ITU-T Video Coding Experts Group Fifth Meeting (ITU-T SG 16 Q.15 Study Period 1997-2000),* Whistler, British Columbia, Canada, July 1998.

[Q15-E-37] CÔTÉ (G.), WENGER (S.): Effects of standard-compliant macroblock intra refresh on rate-distortion performance, *Contribution Q15-E-37 to ITU-T Video Coding Experts Group Fifth Meeting (ITU-T SG16 Q.15 Study Period 1997-2000)*, Whistler, British Columbia, Canada, July 1998.

[Q15-G-22] SONG (H.), KIM (J.), KUO (C.-C.J.): Performance analysis of real-time encoding frame rate control proposal, *Contribution Q15-G-22 to ITU-T Video Coding Experts Group Seventh Meeting (ITU-T SG16 Q.15 Study Period 1997-2000)*, Monterey, California, USA, Feb. 1999.

[Q15-H-24] WIEGAND (T.), FÄRBER (N.), GIROD (B.): Error-Resilient Video Transmission Using Long-Term Memory Motion Compensation, *Contribution Q15-H-24 to ITU-T Video Coding Experts Group Eighth Meeting (ITU-T SG16 Q.15 Study Period 1997-2000)*, Berlin, Germany, Aug. 1999.

[Q15-I-26] HANNUKSELA (M.), LEDISCHKE (M.), ZHANG (J.): Results from Error Resilient Header Repetition Core Experiment, *Contribution Q15-I-26 to ITU-T Video Coding Experts Group Ninth Meeting (ITU-T SG16 Q.15 Study Period 1997-2000)*, Red Bank, New Jersey, USA, Oct. 1999.

[Q15-I-60] WENGER (S.): Common Conditions for Video Performance Evaluation in H.324/M error-prone systems, *Contribution Q15-I-60 to ITU-T Video Coding Experts Group Ninth Meeting (ITU-T SG16 Q.15 Study Period 1997-2000)*, Red Bank, New Jersey, USA, Oct. 1999.

[Q15-J-62] HANNUKSELA (M.): Picture header recovery for H.263 Test Model for H.324/M Use, *Contribution Q15-J-62 to ITU-T Video Coding Experts Group Tenth Meeting (ITU-T SG16 Q.15 Study Period 1997-2000)*, Osaka, Japan, May 2000.

[RFC 1889] SCHULZRINNE (H.), CASNER (S.), FREDERICK (R.), JACOBSON (V.): RTP: A Transport Protocol for Real-time Applications, *IETF RFC 1889*, Jan. 1996.

[RFC 2429] BORMANN (C.), CLINE (L.), DEISHER (G.), GARDOS (T.), MACIOCCO (C.), NEWELL (D.), OTT (J.), SULLIVAN (G.), WENGER (S.), ZHU (C.): RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+), *IETF RFC 2429*, Oct. 1998.

[SG88] SHOHAM (Y.), GERSHO (A.): Efficient Bit Allocation for an Arbitrary Set of Quantizers, in *IEEE Trans. On Acoustics, Speech, and Signal Processing*, Vol. 36, No. 9, pp. 1445-453, Sept. 1988.

[WZG99] WIEGAND (T.), ZHANG (X.), GIROD (B.): Long-Term Memory Motion-Compensated Prediction, in *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, No. 1, pp. 70-84, Feb. 1999.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2002