

I n t e r n a t i o n a l T e l e c o m m u n i c a t i o n U n i o n

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

P.1202.1

(10/2012)

SERIES P: TERMINALS AND SUBJECTIVE AND
OBJECTIVE ASSESSMENT METHODS

Models and tools for quality assessment of streamed
media

Parametric non-intrusive bitstream assessment of video media streaming quality – Lower resolution application area

Recommendation ITU-T P.1202.1



International
Telecommunication
Union

Document Sharing Hub

ITU-T P-SERIES RECOMMENDATIONS

TERMINALS AND SUBJECTIVE AND OBJECTIVE ASSESSMENT METHODS

Vocabulary and effects of transmission parameters on customer opinion of transmission quality	Series	P.10
Voice terminal characteristics	Series	P.30
		P.300
Reference systems	Series	P.40
Objective measuring apparatus	Series	P.50
		P.500
Objective electro-acoustical measurements	Series	P.60
Measurements related to speech loudness	Series	P.70
Methods for objective and subjective assessment of speech quality	Series	P.80
		P.800
Audiovisual quality in multimedia services	Series	P.900
Transmission performance and QoS aspects of IP end-points	Series	P.1000
Communications involving vehicles	Series	P.1100
Models and tools for quality assessment of streamed media	Series	P.1200
Telemeeting assessment	Series	P.1300
Statistical analysis, evaluation and reporting guidelines of quality measurements	Series	P.1400

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



Recommendation ITU-T P.1202.1

Parametric non-intrusive bitstream assessment of video media streaming quality – Lower resolution application area

Summary

Recommendation ITU-T P.1202.1 specifies the algorithmic model for the lower resolution (LR) application area of ITU-T P.1202. The ITU-T P.1202 series of Recommendations specifies models for monitoring the video quality of IP-based video services based on packet-header and bitstream information. The lower resolution application area of the ITU-T P.1202.1 part of ITU-T P.1202 can be applied to the monitoring of the performance and quality of experience (QoE) of video services such as mobile TV.

This Recommendation includes an electronic attachment with test vectors for testing compliance.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T P.1202.1	2012-10-14	12

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). 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.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

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, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2013

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

		Page
1	Scope	1
2	References.....	1
3	Definitions	1
4	Abbreviations and acronyms	2
5	Conventions	2
6	Model description	2
6.1	Extraction of basic parameters	5
6.2	Aggregation of basic parameters into internal picture level parameters	11
6.3	Aggregation of parameters into model parameters	35
6.4	Quality estimation model	41
7	Assumptions	49
8	Model compliance	49
8.1	Test vectors for extraction of basic parameters	50
8.2	Test vectors for aggregation of basic parameters into internal picture level parameters.....	50
8.3	Test vectors for aggregation of parameters into model parameters	50
8.4	Test vectors for quality estimation model	51

Electronic attachment: Test vectors for testing compliance

Downloaded from ITU-T P.1202.1 2012

Recommendation ITU-T P.1202.1

Parametric non-intrusive bitstream assessment of video media streaming quality – Lower resolution application area

1 Scope

This Recommendation¹ describes an algorithmic model for parametric non-intrusive bitstream assessment of video media streaming quality – lower resolution (LR) application area. This algorithmic model can be applied to the monitoring of the video quality of IP-based video services based on information extracted from the video bitstream encoded with the ITU-T H.264/AVC baseline video codec. The model is intended especially for the lower resolution application area, including such services as mobile TV. The ITU-T P.1202.1 model is a no-reference (i.e., non-intrusive) model which operates by analysing information in the video bitstream without fully decoding the bitstream. Further client specific information, such as concealment type and rebuffering duration, is provided to the algorithm out-of-band, for example in the form of stream-specific side information. As output, the model algorithm provides an estimate of the video quality in terms of the five-point absolute category rating (ACR) mean opinion score (MOS) scale defined in [ITU-T P.910]. This Recommendation describes the algorithm that takes an elementary stream encapsulated in RTP/UDP/IP as input, where information from RTP headers is used to identify lost pictures.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- [ITU-T H.264] Recommendation ITU-T H.264 (2012), *Advanced video coding for generic audiovisual services*.
- [ITU-T P.910] Recommendation ITU-T P.910 (2008), *Subjective video quality assessment methods for multimedia applications*.
- [ITU-T P.1202] Recommendation ITU-T P.1202 (2012), *Parametric non-intrusive bitstream assessment of video media streaming quality*.
- [IETF RFC 3550] IETF RFC 3550 (2003), *RTP: A Transport Protocol for Real-Time Applications*.
- [IETF RFC 3984] IETF RFC 3984 (2005), *RTP Payload Format for H.264 Video*.
- [IETF RFC 4566] IETF RFC 4566 (2006), *SDP: Session Description Protocol*.

3 Definitions

None.

¹ This Recommendation includes an electronic attachment with test vectors for testing compliance.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ACR	Absolute Category Rating
AVC	Advanced Video Coding
DCT	Discrete Cosine Transform
EC MB	Error Concealed Macroblock
GOP	Group Of Pictures
IP	Internet Protocol
MB	Macroblock
MOS	Mean Opinion Score
NAL	Network Abstraction Layer
PLC	Packet Loss Concealment
QP	Quantization Parameter
RTP	Real Time Protocol
SDP	Session Description Protocol
SPS	Sequence Parameter Set
UDP	User Datagram Protocol

5 Conventions

None.

6 Model description

This clause describes the ITU-T P.1202.1 model and how it is implemented. The block diagram for the model is depicted in Figure 6-1. The model takes an ITU-T H.264/AVC baseline encoded video bitstream and side information (error concealment type, rebuffering information, etc.) as input, extracts parameters, also referred to as features, and aggregates them into model parameters which are used to calculate an estimated video quality MOS for the sequence.

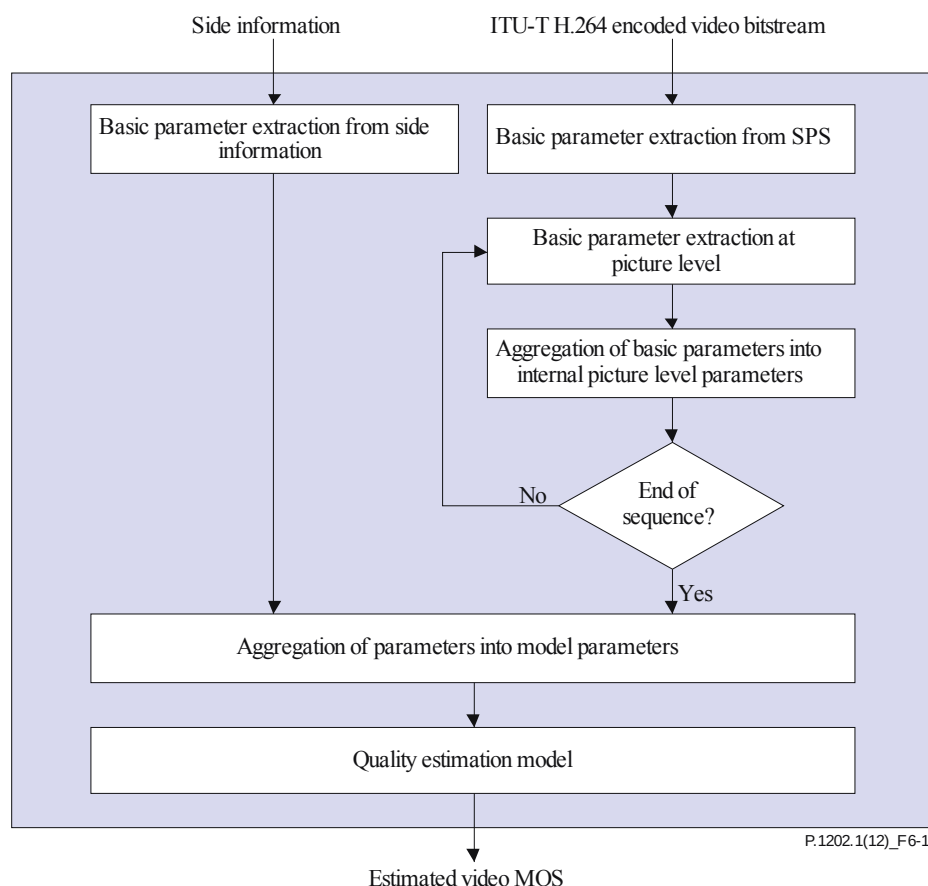


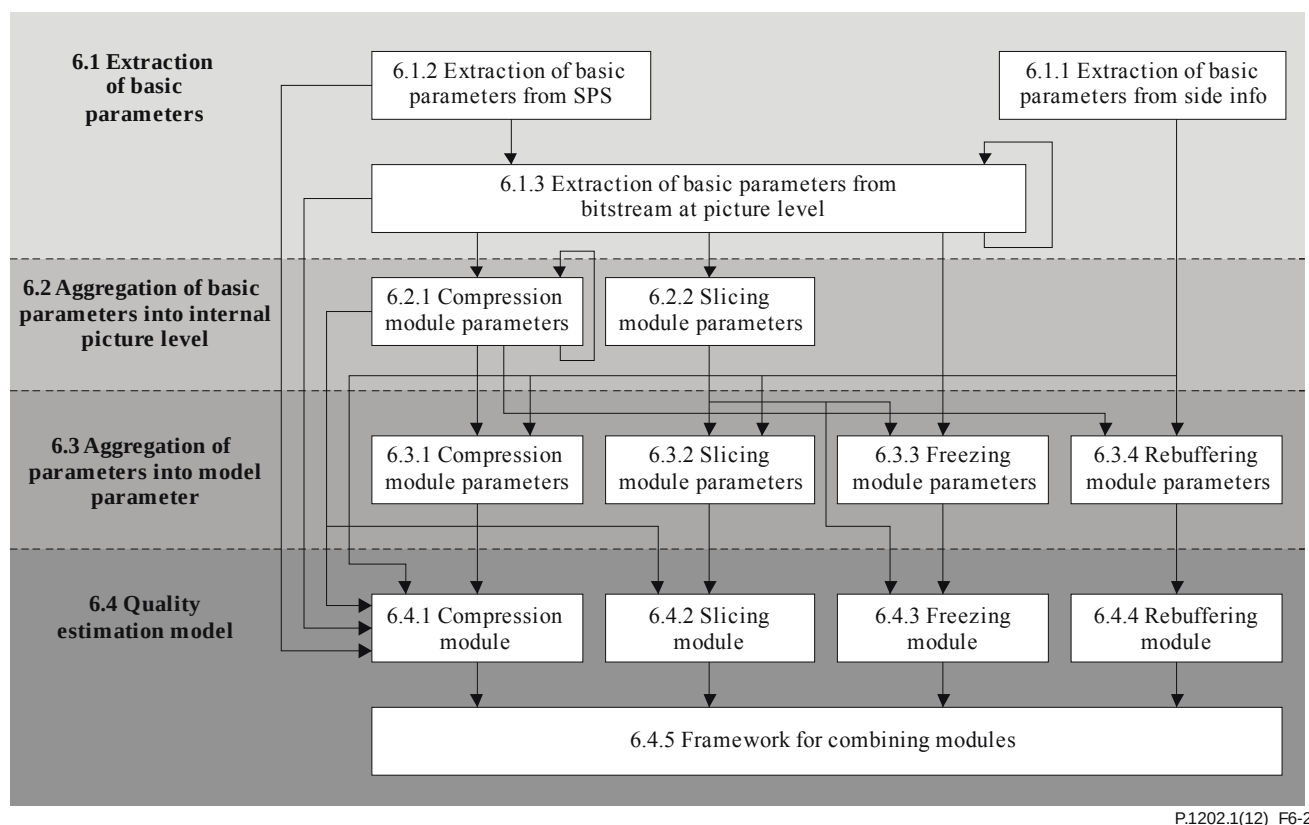
Figure 6-1 – Block diagram for ITU-T P.1202.1

The model description is organized according to the flow of parameters in the model and to which degradation module the parameters belong. Clause 6.1 describes the extraction of basic parameters from the video bitstream and side information. In clause 6.2 the algorithms for aggregating the basic parameters into internal parameters at picture level are described. The words picture and frame are used interchangeably throughout this Recommendation. The aggregation of parameters into model parameters is described in clause 6.3, and finally the quality estimation model is described in clause 6.4.

Four different types of degradations are detected by the ITU-T P.1202.1 model; compression artefacts, slicing artefacts, freezing artefacts and rebuffering artefacts. Compression artefacts are introduced due to lossy compression of the encoding process. Slicing artefacts are introduced when packet losses are concealed using a packet loss concealment (PLC) scheme trying to repair erroneous frames. Freezing artefacts are introduced when the PLC scheme of the receiver replaces the erroneous frames (either due to packet loss or error propagation) with the previous error free frame until a decoded picture without errors has been received. Since the erroneous frames are not displayed, this type of artefact is also referred to as freezing with skipping. Rebuffering artefacts come from rebuffering events at the client side, which could be a result of video data arriving late. Usually, rebuffering events are indicated to the viewer, e.g., in the form of a spinning wheel. This is also referred to as freezing without skipping.

For each clause a set of input and output parameters that are used by the algorithms are defined and described. Model coefficients are also given if it is being used by the algorithms.

The flow chart in Figure 6-2 describes the flow of parameters in the model and in what clause each part can be found.



P.1202.1(12)_F6-2

Figure 6-2 – Flow of parameters in the model

The algorithmic descriptions of the model are described using C-style pseudocode, leaving out variable declarations, row-ending semi-colons, etc., for better readability. The variables in the algorithm descriptions are prefixed according to what variable type they pertain to, e.g., *i_nbr_mbs*. The prefixes have the following meaning:

- b – boolean
- i – integer
- d – double, i.e., floating point value in double precision
- s – string
- p – struct or object holding data

The following static code words are used for the macroblock data in the pseudocode of the algorithmic descriptions in this clause:

INTER16x16_MB	The macroblock is an inter macroblock divided into macroblock partitions of size 16x16. It is the same as P_L0_16x16 in [ITU-T H.264].
INTER16x8_MB	The macroblock is an inter macroblock divided into macroblock partitions of size 16x8. It is the same as P_L0_L0_16x8 in [ITU-T H.264].
INTER8x16_MB	The macroblock is an inter macroblock divided into macroblock partitions of size 8x16. It is the same as P_L0_L0_8x16 in [ITU-T H.264].

INTER8x8_MB	The macroblock is an inter macroblock divided into macroblock partitions of size 8x8 with possibly different reference pictures for each macroblock partition. It is the same as P_8x8 in [ITU-T H.264].
INTER8x8REF0_MB	The macroblock is an inter macroblock divided into macroblock partitions of size 8x8 where all macroblock partitions uses the first reference picture in the list. It is the same as P_8x8ref0 in [ITU-T H.264].
SKIPPED_MB	The macroblock is of skip type.
INTRA_MB	The macroblock is an intra-macroblock
INTRA16x16_MB	The macroblock is an intra-macroblock with 16x16 intra prediction mode. It is the same as Intra_16x16 in [ITU-T H.264] with any setting of Intra16x16PredMode, CodedBlockPatternChroma and CodedBlockPatternLuma.
LOST	The macroblock has been lost
UNKNOWN	The type of the macroblock is unknown

6.1 Extraction of basic parameters

The extraction of basic parameters is done from side information; from sequence parameter set (SPS) network abstraction layer (NAL) units in the video bitstream, or from the picture level in the video bitstream. If an SPS NAL unit does not exist in the video bitstream, the output in clause 6.1.2.2 can be derived from session description protocol (SDP) data or from other side information.

6.1.1 Extraction of basic parameters from side information

The basic parameter extraction from side information has the following input and output parameters:

Input = [side_information]

Output = [d_avg_frame_rate,
d_rebuffering_duration,
s_video_PLC_mode]

6.1.1.1 Input parameter description

side_information Side information containing necessary information for calculating the model score correctly.

6.1.1.2 Output parameter description

d_avg_frame_rate Average number of frames per second for the sequence. *d_avg_frame_rate* should be taken from *side_information* if available, otherwise *d_avg_frame_rate* should be derived from the timestamp in the RTP header.

d_rebuffering_duration

Total combined duration of rebuffering in seconds for the sequence

s_video_PLC_mode

Packet loss concealment method employed in video decoder. The value is "SLICING" or "FREEZING".

6.1.2 Extraction of basic parameters from SPS

The sequence parameter set (SPS) contains information necessary to properly decode a sequence. The basic parameter extraction from SPS has the following input and output parameters:

Input = [bitstream]

Output = [i_nbr_mb_ver,
i_nbr_mbs,
b_gaps_in_frame_num_value_allowed_flag]

6.1.2.1 Input parameter description

bitstream A video bitstream encoded using the ITU-T H.264/AVC baseline profile.

6.1.2.2 Output parameter description

i_nbr_mb_ver Number of macroblocks in vertical direction for a picture.

i_nbr_mbs Number of macroblocks in a picture calculated as $i_nbr_mbs = i_nbr_mb_hor \times i_nbr_mb_ver$, where $i_nbr_mb_hor$ is the number of macroblocks in horizontal direction for a picture.

b_gaps_in_frame_num_value_allowed_flag

Flag indicating whether a gap in the frame_num value is allowed. If it is not allowed and a gap is present, it means that one or more pictures have been lost.

6.1.3 Extraction of basic parameters from bitstream at picture level

The extraction of basic parameters from the bitstream at picture level takes part both on RTP level and at picture level in the elementary stream. The extraction of basic parameters from bitstream at picture level has the following input and output parameters:

Input = [b_first_picture,
RTP_bitstream,
i_nbr_mbs,
p_mb_data_array[]]

struct MB_data

```
{  
    i_dct_coef_ac[][],  
    i_dct_coef_dc[],  
    i_cbp,  
    i_mb_type,  
    i_qp,  
    i_mv_x[],  
    i_mv_y[],  
    i_ref_idx,  
    i_sub_mb_type  
}
```

struct MB_data p_mb_data_array[i_nbr_mbs]

Output = [i_curr_mb_type_map[],
i_curr_qp_map[],
i_curr_avg_mv_x_map[],
i_curr_avg_mv_y_map[],
i_max_nbr_used_ref_pics,
b_sub_mb_type_inter8x4_or_inter4x8_exists,
b_sub_mb_type_inter4x4_exists,
i_frame_num,
i_displayorder,

```

i_received_packets,
i_lostpackets,
i_received_bytes,
i_frame_type,
b_reflost0,
i_lostframegap]

```

6.1.3.1 Input parameter description

b_first_picture	Boolean indicating if the current picture is the first picture in the sequence. For the first picture this variable should be TRUE, for all other pictures FALSE.
RTP_bitstream	The video bitstream encapsulated in RTP packets.
i_nbr_mbs	Number of macroblocks in a picture.
p_mb_data_array[]	Array of structures of type "struct MB_data" containing macroblock data. The size of the array is <i>i_nbr_mbs</i> . The macroblock data in the structs contain the entries described below.

The following parameters of *p_mb_data_array[]* are obtained for each MB in a picture.

i_dct_coef_ac[][]	AC components of DCT coefficients of prediction residual. Its dimension is 16x16. First dimension corresponds to 16 4x4 sized-blocks; second dimension corresponds to 16 DCT coefficients of 4x4 DCT transform.
i_dct_coef_dc[]	DC components of DCT coefficients of prediction residual. Its dimension is 16, corresponding to Hadamard Transform coefficients of DC components of DCT transform of each 4x4 size-block.
i_cbp	It has the same meaning as the cbp defined in ITU-T H.264 syntax.
i_mb_type	Macroblock type
i_qp	QP for the macroblock
i_mv[]	Array of horizontal motion vectors at sub-macroblock level. The size of the array is 16.
i_mvy[]	Array of vertical motion vectors at sub-macroblock level. The size of the array is 16.
i_ref_idx	Index of the reference picture for the first macroblock partition in the macroblock.
i_sub_mb_type	Sub-macroblock type for the first of the four possible sub-macroblock in the macroblock.

6.1.3.2 Output parameter description

The following output parameters are obtained for each parsed picture. Parsed pictures include completely received pictures and partially received pictures.

i_curr_mb_type_map[]	Array containing macroblock types for the macroblocks in the current picture. The size of the array is <i>i_nbr_mbs</i> .
i_curr_qp_map[]	Array containing QP values for the macroblocks in the current picture. The size of the array is <i>i_nbr_mbs</i> . Same as <i>pm_data_array[]</i> .i_qp for the current picture.
i_curr_avg_mv[]	Array containing averaged horizontal motion vector values for the macroblocks in the current picture. The size of the array is <i>i_nbr_mbs</i> .



i_curr_avg_mvy_map[]	Array containing averaged vertical motion vector values for the macroblocks in the current picture. The size of the array is <i>i_nbr_mbs</i> .
i_max_nbr_used_ref_pics	The number of actually used reference pictures detected in the bitstream so far. This parameter is detected by checking for the maximum reference picture index for the first macroblock partition in each received macroblock. This parameter is updated for each picture.
b_sub_mb_type_inter8x4_or_inter4x8_exists	Boolean indicating if a sub-macroblock of type inter 8x4 or inter 4x8 exists in the bitstream so far. This parameter is updated for each picture.
b_sub_mb_type_inter4x4_exists	Boolean indicating if a sub-macroblock of type inter 4x4 exists in the bitstream so far. This parameter updated for each picture.
i_frame_num	The frame number counter in the ITU-T H.264 video bitstream as defined in [ITU-T H.264].
i_displayorder	Display order of a picture in the bitstream corresponding to the picture number. It is obtained from the RTP timestamp for the ITU-T H.264/AVC bitstream encoded with baseline profile, where the display order and encoding order are the same.
i_received_packets	The number of received RTP packets belonging to a picture.
i_lostpackets	The number of lost RTP packets belonging to a picture. The RTP packets lost between the start of current picture and the end of immediately previous picture are counted into current picture's <i>i_lostpackets</i> .
i_received_bytes	The number of received bytes belonging to a picture in the elementary stream.
i_frame_type	The frame type takes one of the following values; 0 = I_frame, 1 = P_frame, 2 = B_frame (not applicable for ITU-T H.264/AVC baseline) or 5 = unknown.
b_reflost0	Boolean indicating whether a reference picture in current pictures' <i>reference_picture_list0</i> is totally lost. Default/initial value is FALSE.
i_lostframegap	The number of consecutively totally lost frames from the latest fully/partly received frame. This can be derived from the gap of RTP timestamp.

6.1.3.3 Algorithms

At this stage, basic parameters are extracted and updated for the current picture. Some of the basic picture level information is obtained at the RTP level including the number of received packets for a picture, the number of lost packets for a picture, the number of received bytes belonging to a picture and the number of consecutively totally lost pictures.

The other basic parameters are obtained at picture level in the elementary stream. An array of structs holding macroblock information, *p_mb_data_array[]*, is used for the data received from the parser for the description in this clause.

Keeping track of lost pictures can be done using timestamp information from the RTP header or be derived from the *frame_num* parameter in the bitstream. The most reliable and straightforward

method is to use the timestamp from the RTP header. How to keep track of lost pictures from RTP timestamp and frame_num is explained below.

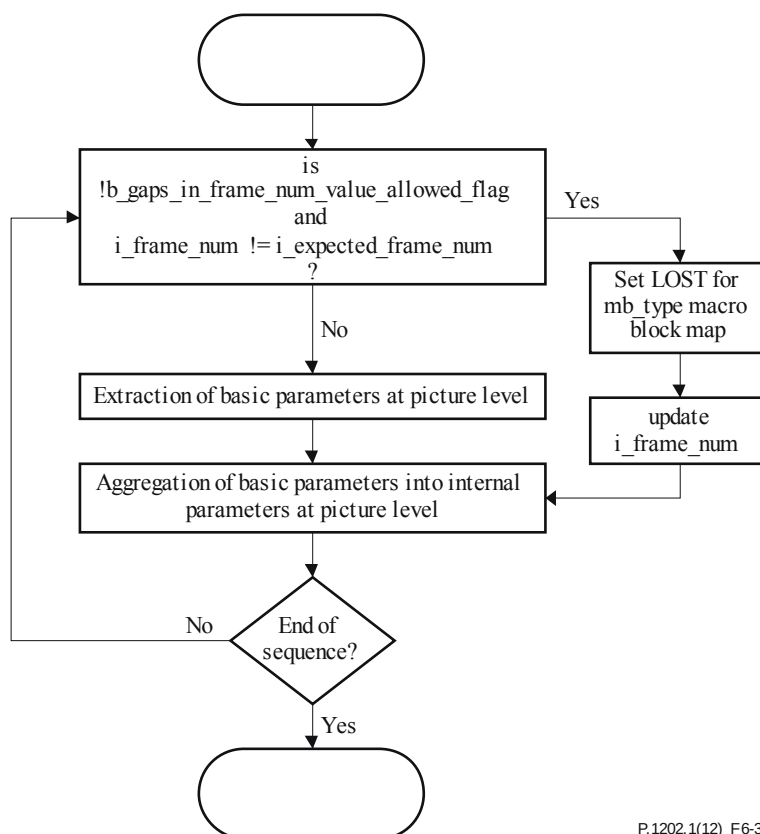
6.1.3.3.1 Keeping track of lost packets and pictures using RTP timestamp

RTP packets having the same RTP timestamp belong to the same picture. The sequence number is defined in the RTP protocol header and it increments by one for each sent RTP packet. The number of lost RTP packets between the first and last received RTP packets belonging to frame i is counted based on the discontinuity of sequence numbers. Similarly, the lost RTP packets between the first received RTP packet of frame i and the last received RTP packet of its immediately previous frame $i-1$ are counted based on the discontinuity of sequence numbers and counted into the lost packets of frame i .

It can be detected if one or more consecutive pictures have been lost by checking if the difference in RTP timestamp for the preceding and following packets of where the loss occurs exceeds the expected RTP timestamp increment by a factor n , where n is an integer equal to or larger than 0. If $n > 1$, then the number of lost pictures for the gap is $n-1$. The expected RTP timestamp increment can be derived from the video frame rate and the resolution of the sample clock.

6.1.3.3.2 Keeping track of lost pictures using i_frame_num

The flow chart below illustrates how lost pictures are detected using the *i_frame_num* parameter extracted from the bitstream. For the first picture the current frame number is the expected frame number. If the expected frame number equals the current frame number, the basic parameters are extracted as described in this clause followed by aggregating the basic parameters into internal parameters as described in clause 6.2. If the current frame is not equal to the expected frame number and the flag *b_gaps_in_frame_num_value_allowed_flag* is not set, a picture has been lost. The LOST state is then updated in all macroblock positions for the map holding the information about macroblock type for the current picture.



P.1202.1(12)_F6-3

Figure 6-3 – Flow chart describing how lost pictures are detected using i_frame_num

The following pseudocode describes the procedure for keeping track of lost frames using `i_frame_num`:

```
while (!EndOfSequence) {
    if (b_first_picture) {
        i_expected_frame_num = i_frame_num
    }

    if (!b_gaps_in_frame_num_value_allowed_flag &&
        i_frame_num != i_expected_frame_num) {

        // Set all macro block types for current picture to LOST
        for (i_mb = 0; i_mb < i_nbr_mbs; i_mb++) {
            i_curr_mb_type_map[i_mb] = LOST
        }

        // update frame number
        i_frame_num = (i_frame_num + 1) % MaxFrameNum

        do aggregate basic parameters into internal parameters at picture level

    } else { // The picture is not lost

        i_expected_frame_num = (i_frame_num + 1) % MaxFrameNum

        do extract basic parameters at picture level

        do aggregate basic parameters into internal parameters at picture level
    }
}
```

6.1.3.3.3 Obtaining basic parameters from the bitstream at picture level

This clause describes how to obtain some of the needed basic parameters from the bitstream at picture level.

Macroblock maps containing macroblock type, QP value and average vertical and horizontal motion vectors for the macroblock are updated and kept in array maps for all macroblocks in the current picture. If a macroblock has been lost the corresponding value the macroblock type map for that macroblock is set to LOST.

Parameters related to the calculation of the encoder complexity level as described in clause 6.4.1 is also extracted at this stage.

The following pseudocode shows how to extract some of the basic parameters at picture level:

```
if (b_first_picture) {
    // Initialize parameters
    i_max_nbr_used_ref_pics = 0
    b_sub_mb_type_inter8x4_or_inter4x8_exists = FALSE
    b_sub_mb_type_inter4x4_exists = FALSE
}

for (i_mb = 0; i_mb < i_nbr_mbs; i_mb++) {
    if (p_mb_data_array[i_mb].mb_type == UNKNOWN) {
        // Set that MB has been lost
        i_curr_mb_type_map[i_mb] = LOST
    } else {
        // Update values
        i_curr_mb_type_map[i_mb] = p_mb_data_array[i_mb].i_mb_type
    }
}
```

```

i_curr_qp_map[i_mb]          = p_mb_data_array[i_mb].i_qp

// Reset mv values
i_curr_avg_mv_x_map[i_mb] = 0
i_curr_avg_mv_y_map[i_mb] = 0

// Traverse sub-macro blocks and set avg MVs for non-intra MBs
for (i_smb = 0; i_smb < 16; i_smb++) {
    if (p_mb_data_array[i_mb].mb_type != INTRA_MB) {
        i_curr_avg_mv_x_map[i_mb] += p_mb_data_array[i_mb].i_mv_x[i_smb]
        i_curr_avg_mv_y_map[i_mb] += p_mb_data_array[i_mb].i_mv_y[i_smb]
    }
}
i_curr_avg_mv_x_map[i_mb] /= 16
i_curr_avg_mv_y_map[i_mb] /= 16

// Check for parameters related to the encoder complexity

// Check for nbr of actually used reference pictures in the first macro
// block partition only
if (i_max_nbr_used_ref_pics < p_mb_data_array[i_mb].i_ref_idx + 1) {
    i_max_nbr_used_ref_pics = p_mb_data_array[i_mb].i_ref_idx + 1
}

// Check for inter 8x4 or inter 4x8 sub macro block type
if (!b_sub_mb_type_inter8x4_or_inter4x8_exists &&
    (p_mb_data_array[i_mb].mb_type == INTER8x8_MB ||
     p_mb_data_array[i_mb].mb_type == INTER8x8REF0_MB) &&
    (p_mb_data_array[i_mb].sub_mb_type == 1 ||
     p_mb_data_array[i_mb].sub_mb_type == 2)) {
    b_sub_mb_type_inter8x4_or_inter4x8_exists = TRUE
}

// Check for inter 4x4 sub macro block type
if (!b_sub_mb_type_inter4x4_exists &&
    (p_mb_data_array[i_mb].mb_type == INTER8x8_MB ||
     p_mb_data_array[i_mb].mb_type == INTER8x8REF0_MB) &&
    p_mb_data_array[i_mb].sub_mb_type == 0) {
    b_sub_mb_type_inter4x4_exists = TRUE
}
}
}

```

6.2 Aggregation of basic parameters into internal picture level parameters

This clause describes how internal picture level parameters are aggregated from the basic parameters extracted from the video bitstream.

6.2.1 Compression module parameters

The aggregation of basic parameters into internal picture level parameters used for the compression module uses the following input and output parameters.

Input = [b_first_picture,
i_nbr_mbs,
i_curr_qp_map[],
i_curr_mb_type_map[],
i_curr_avg_mv_x_map[],
i_curr_avg_mv_y_map[]]

Output = [d_avg_qp_curr_pic,
d_avg_qp_prev_pic,
d_min_avg_pic_qp,
d_max_avg_pic_qp,
d_sum_avg_pic_qp,
d_sum_avg_pic_mv,

```

i_prev_mb_type_map[],
b_curr_pic_is_intra_like,
i_nbr_intra_pics,
i_nbr_inter_pics,
b_has_intra_pic_flicker,
i_nbr_received_pics,
i_nbr_lost_pics,
i_loss_penalty_map[],
d_sum_max_nbr_partitions]

```

6.2.1.1 Input parameter description

b_first_picture	See clause 6.1.3.1
i_nbr_mbs	See clause 6.1.2.2
i_curr_qp_map[]	See clause 6.1.3.2
i_curr_mb_type_map[]	See clause 6.1.3.2
i_curr_avg_mv_x_map[]	See clause 6.1.3.2
i_curr_avg_mv_y_map[]	See clause 6.1.3.2

6.2.1.2 Output parameter description

d_avg_qp_curr_pic	The average QP value for all correctly received macroblocks in the current picture.
d_avg_qp_prev_pic	The average QP value for all correctly received macroblocks in the previous picture.
d_min_avg_pic_qp	The QP value over all macroblocks are averaged for each picture. This parameter is the value for the picture with the smallest value of all pictures received so far.
d_max_avg_pic_qp	The QP value over all macroblocks are averaged for each picture. This parameter is the value for the picture with the largest value of all pictures received so far.
d_sum_avg_pic_qp	The sum of all average QP per picture for all pictures received so far.
d_sum_avg_pic_mv	Average motion per picture summed up for all received pictures so far. Used for calculating average motion for the whole sequence.
i_prev_mb_type_map[]	Array containing macroblock types for the macroblocks in the previous picture. The size of the array is <i>i_nbr_mbs</i> .
b_curr_pic_is_intra_like	Indicates if the current picture is considered intra-like.
i_nbr_intra_pics	The number of intra pictures received so far.
i_nbr_inter_pics	The number of inter pictures received so far.
b_has_intra_pic_flicker	Boolean indicating if there has been a detected intra picture flicker in the received bitstream up till the current picture.
i_nbr_received_pics	The number of pictures received so far.
i_nbr_lost_pics	The number of lost pictures so far in the bitstream.
i_loss_penalty_map[]	An array keeping track of if an error is propagating for each macroblock in the picture. The size of the array is <i>i_nbr_mbs</i> . The initial values in the array are all 0.

d_sum_max_nbr_partitions

The maximum number of macroblock partitions in a macroblock for a picture having a propagating error, summarized over all pictures received so far.

6.2.1.3 Algorithms

6.2.1.3.1 Aggregation of basic parameters into internal parameters at picture level

This clause describes how the basic parameters are aggregated into internal parameters at picture level. If it is the first picture, all parameters that are updated for each picture are first initialized. The following internal parameters are aggregated at picture level; average motion perception, average, min and max QP value, number of inter and intra pictures, detection of intra-picture flicker and maximum number of macroblock partitions used for the picture if an error is propagating. All of this is further described in the pseudocode below and the following clauses.

```
if (b_first_picture) {  
  
    // Initialize parameters before first picture  
    d_avg_qp_prev_pic          = 0  
    d_avg_qp_curr_pic          = 0  
    d_sum_avg_pic_mv           = 0  
    d_min_avg_pic_qp           = INT_MAX  
    d_max_avg_pic_qp           = 0  
    d_sum_avg_pic_qp           = 0  
    d_sum_max_nbr_partitions   = 0  
    i_nbr_intra_pics           = 0  
    i_nbr_inter_pics           = 0  
    i_nbr_received_pics        = 0  
    i_nbr_lost_pics            = 0  
    b_curr_pic_is_intra_like    = 0  
    b_has_intra_pic_flicker     = 0  
}  
  
// Initialize internal picture level counters and sums  
i_qp_pic_sum                  = 0  
i_nbr_lost_mbs                = 0  
i_nbr_intra_mb                = 0  
i_motion_sum                  = 0  
d_nbr_mbs_that_may_have_mv    = 0  
d_max_nbr_partitions          = 0  
  
for (i_mb = 0; i_mb < i_nbr_mbs; i_mb++) {  
  
    if (i_curr_mb_type_map[i_mb] == LOST) {  
        i_nbr_lost_mbs++  
    } else {  
        i_qp_pic_sum += i_curr_qp_map[i_mb]  
    }  
  
    if (i_curr_mb_type_map[i_mb] == INTRA_MB) {  
        i_nbr_intra_mb++  
    }  
  
    // Calculate average motion perception  
    if ((i_curr_mb_type_map[i_mb] != INTRA_MB) &&  
        (i_curr_mb_type_map[i_mb] != LOST)) {  
        i_motion_sum += (abs(i_curr_avg_mv_x_map[i_mb]) +  
                        abs(i_curr_avg_mv_y_map[i_mb])) / 2  
        d_nbr_mbs_that_may_have_mv++  
    }  
}
```