



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

P.1202.2

(05/2013)

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 – higher
resolution application area**

CAUTION !

PREPUBLISHED RECOMMENDATION

This prepublication is an unedited version of a recently approved Recommendation. It will be replaced by the published version after editing. Therefore, there will be differences between this prepublication and the published version.

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/had not] 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.

Recommendation ITU-T P.1202.2

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

Summary

Recommendation ITU-T P.1202.2 specifies the algorithmic model for the higher resolution (HR) application area of ITU-T P.1202. The ITU-T P.1202 series of documents specifies models for monitoring the video quality of IP-based video services based on packet-header and bitstream information. The higher resolution application area of ITU-T P.1202.2 part of ITU-T P.1202 can be applied to the monitoring of performance and quality of experience (QoE) of video services such as IPTV, and has two modes: Mode 1, where the video bitstream are parsed and not decoded into pixels, and mode 2, where the video bitstream are fully decoded into pixels for analyzing.

Table of Contents

1	Scope	3
2	References and abbreviations	3
2.1	References	3
2.2	Abbreviations	3
3	Model mode 1 description	4
3.1	Extraction of basic parameters	7
3.1.1	Extraction of basic parameters from side information	7
3.1.2	Extraction of basic parameters from SPS/PPS	8
3.1.3	Extraction of basic parameters from bitstream at picture level	8
3.2	Aggregation of basic parameters into internal picture level parameters	11
3.2.1	Compression module parameters	11
3.2.2	Slicing module parameters	14
3.2.3	Freezing module parameters	34
3.3	Aggregation of parameters into model parameters	35
3.3.1	Compression module parameters	35
3.3.2	Slicing module parameters	37
3.3.3	Freezing module parameters	43
3.4	Quality estimation model	43
3.4.1	Compression module	44
3.4.2	Slicing module	45
3.4.3	Freezing module	46
3.4.4	Framework for combining modules	47
4	Model mode 2 description	49
4.1	Feature extraction and aggregation	54
4.1.1	Compression impairment	54
4.1.2	Freezing impairment	55
4.1.3	Slicing impairment	56
4.2	Coefficient setting	61
5	Assumptions	62
6	Model compliance	62
6.1	Test vector for quality estimation model of mode 1	63
6.2	Test vector for quality estimation model of mode 2	64
7	Bibliography	64

1 Scope

Recommendation ITU-T P.1202.2 describes parametric non-intrusive bitstream assessment of video media streaming quality - higher resolution application area (HR), for monitoring the video quality of IP-based video services based on information extracted from the video bitstream encoded with the H.264/AVC baseline video codec. The model are intended especially for the higher resolution application area, including services as IPTV. Recommendation ITU-T P.1202.2 consists of two models corresponding to two modes: mode 1 and mode 2, which both are no-reference (i.e. non-intrusive) models. Mode 1 refers to as parsing mode, the model operates by analysing information in the video bitstream without fully decoding the bitstream (i.e. no pixel information is used) for MOS estimation, and Mode 2 refers to as full decoding mode, in addition to the bitstream information which mode 1 uses, the model can also decode parts or all of the video bitstream (i.e. pixel information is used) for MOS estimation. Further client specific information, such as concealment type, 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 5-point absolute category rating (ACR) mean opinion score (MOS) scale defined in ITU-T P.910.

2 References and abbreviations

2.1 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 P.1202] Recommendation ITU-T P.1202 (2012), Parametric non-intrusive bitstream assessment of video media streaming quality

[ITU-T P.910] Recommendation ITU-T P.910 (2008), Subjective video quality assessment methods for multimedia applications

[ITU-T H.264] Recommendation ITU-T H.264 (2012) Advanced video coding for generic audiovisual services

[ITU-T Rec. H.222.0, ISO/IEC 13818-1] Rec. ITU-T H.222.0 | ISO/IEC 13818-1, Information technology -- Generic coding of moving pictures and associated audio information: Systems

[IETF RFC 3550] IETF RFC 3550 (2003), RTP: A Transport Protocol for Real-Time Applications

[IETF RFC 4566] IETF RFC 4566 (2006), SDP: Session Description Protocol

[IETF RFC 2250] IETF RFC 2250 (1998), RTP Payload Format for MPEG1/MPEG2 Video

2.2 Abbreviations

ACR Absolute Category Rating

EC MB Error Concealed Macro Block

GOP	Group Of Pictures
H.264/AVC	H.264 Advanced Video Coding
MB	Macro Block
MOS	Mean Opinion Score
MV	Motion Vector
NAL	Network Abstraction Layer
PLC	Packet Loss Concealment
SPS	Sequence Parameter Set
PPS	Picture Parameter Set
POC	Picture Order Count

3 Model mode 1 description

This section describes the ITU-T Recommendation P.1202.2 model mode 1 and how it is implemented. The block diagram for the model is depicted in figure 3-1. The model takes an H.264/AVC encoded video bitstream and side information (error concealment type, 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.

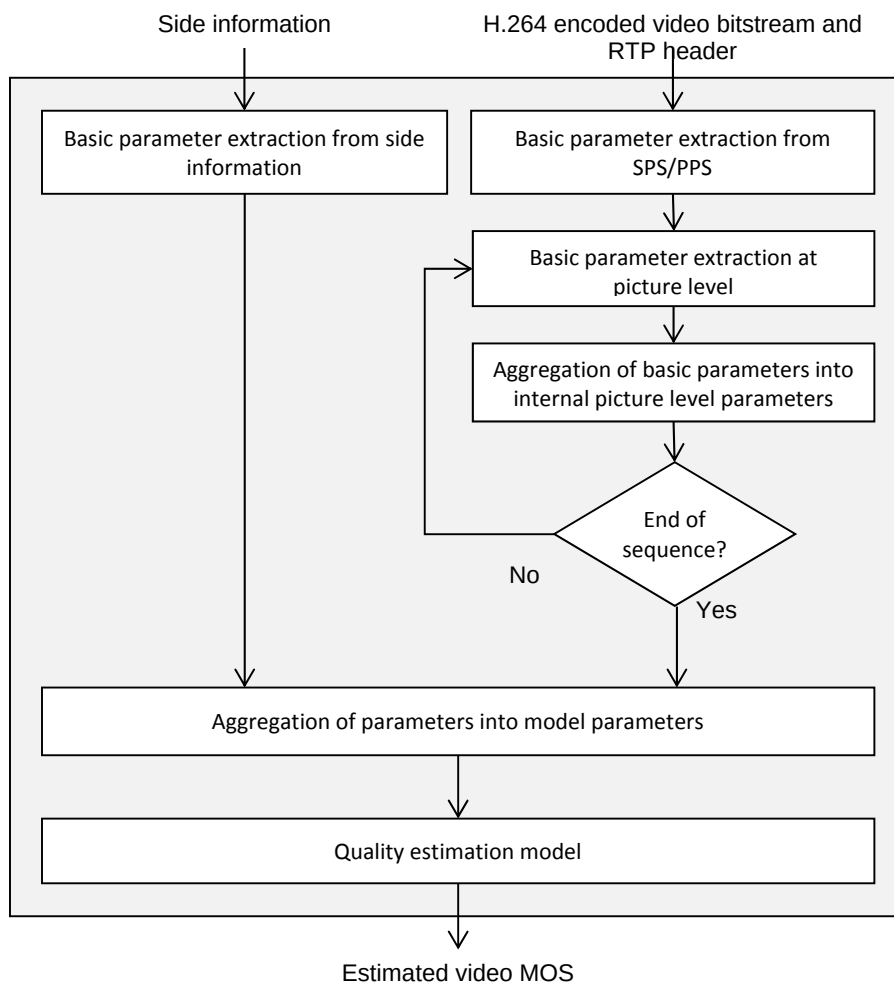


Figure 3-1 – Block diagram for P.1202.2

The model description is organised according to the flow of parameters in the model and to what degradation module the parameters belong to. Subsection 3.1 describes the extraction of basic parameters from the video bitstream and side information. In the second subsection, 3.2, 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 subsection 3.3, and finally the quality estimation model is described in subsection 3.4.

Three different types of degradations are detected by the Recommendation ITU-T P.1202.2 model; compression artifacts, slicing artifacts and freezing artifacts. Compression artifacts are introduced due to lossy compression of the encoding process. Slicing artifacts are introduced when packet losses are concealed using a packet loss concealment (PLC) scheme trying to repair erroneous frames. Freezing artifacts 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 artifact is also referred to as freezing with skipping. The word *error free* means *no packet losses occurred* throughout the Recommendation.

For each subsection 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 3-2 describes the flow of parameters in the model and in what subsection each part can be found. Please note that the dotted line is used between section 3.4.1 and 3.4.5, and it means that when artifact is only caused by video compression, the output of section 3.4.1 (*d_compression_quality_value*) is output directly as the overall video quality score. Otherwise, when there is one or more other artifact type, for example, artifact caused by packet losses, the framework module is used to consider degradations caused by various artifact types to generate a total video quality score, i.e. the output of section 3.4.1 (*d_compression_quality_value*), 3.4.2 (*d_slicing_artifact_value*) and 3.4.3 (*d_freezing_artifact_value*) will go into section 3.4.5.

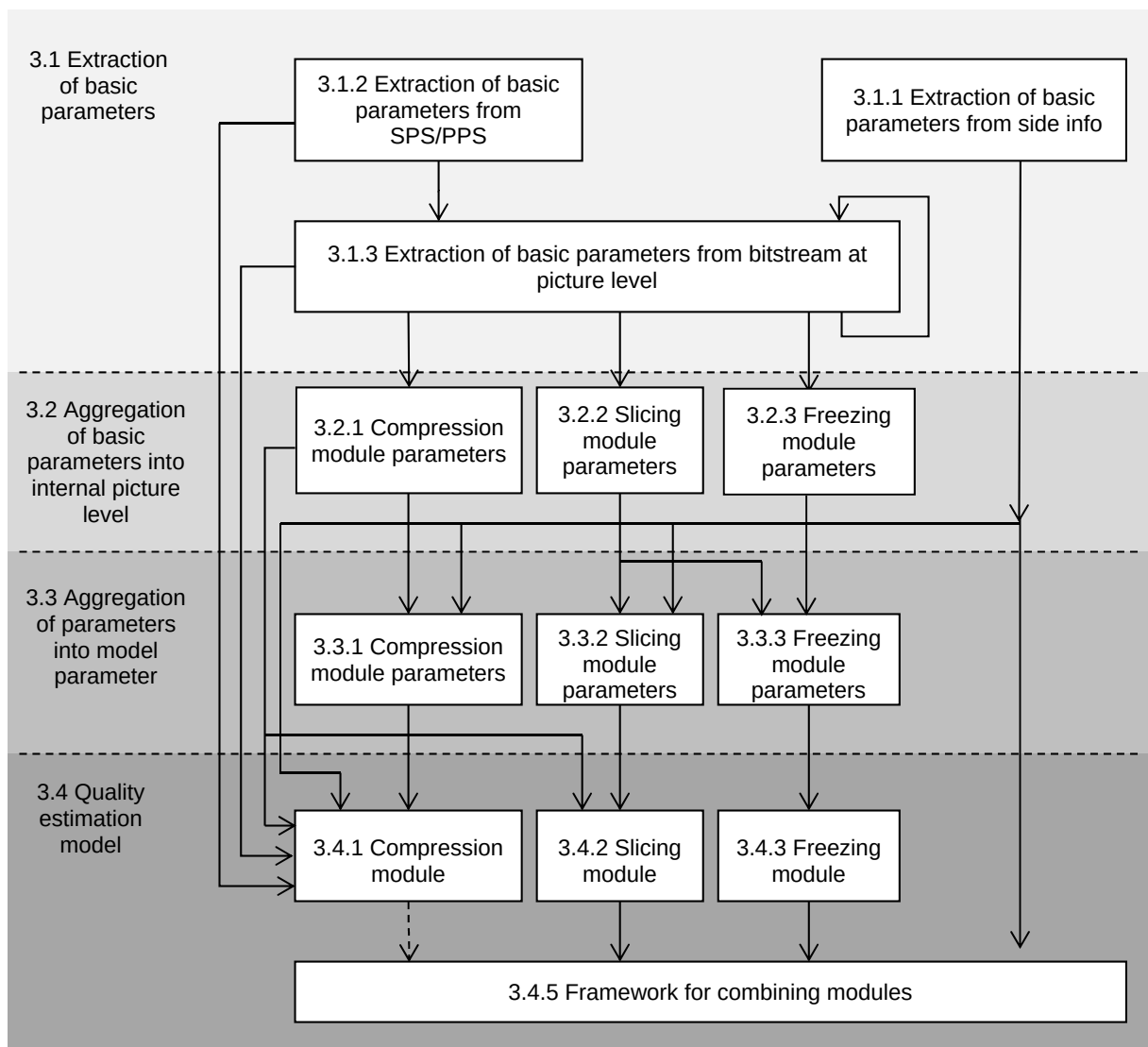


Figure 3-2 - Flow of parameters in the model

The algorithmic descriptions of the model are described using pseudocode. 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
- c – unsigned char

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

The following static code words are used for the macro block data in the pseudocode of the algorithmic descriptions in this section:

INTRA_MB	The macro block is an intra macro block
INTRA16x16_MB	The macro block is an intra macro block with 16x16 intra prediction mode. It is the same as Intra_16x16 in the H.264 Recommendation with any setting of Intra16x16PredMode, CodedBlockPatternChroma and CodedBlockPatternLuma.
MB_8X8_TRANSFRM	The prediction residual signal of the macro block is encoded with 8x8 Transform as described in H.264 Recommendation.
UNKNOWN	The type of the macro block is unknown

3.1 Extraction of basic parameters

The extraction of basic parameters is done from side information; from SPS (Sequence Parameter Set) NAL (Network Abstraction Layer) and PPS (Picture Parameter Set) NAL units in the video bitstream, or from the picture level in the video bitstream.

3.1.1 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 = [f_fps,
s_video_PLC_mode
i_numofslices]

3.1.1.1 Input parameter description

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

3.1.1.2 Output parameter description

f_fps The number of frames per second for the sequence. *f_fps* should be taken from *side_information* if available, or *f_fps* should be taken from transport packet header.

s_video_PLC_mode Packet loss concealment method employed in a video decoder. The value is “SLICING” or “FREEZING” or “N/A”.

i_numofslices The number of slice(s) in a frame. This can be taken from *side_information* if available, otherwise *i_numofslices* can be derived from H.264 bitstream.

3.1.2 Extraction of basic parameters from SPS/PPS

Input = [bitstream]

Output = [i_nbr_mb_ver,
i_nbr_mb_hor,
i_nbr_mbs,
i_pic_init_qp_minus26]

3.1.2.1 Input parameter description

bitstream A video bitstream encoded using the H.264/AVC.

3.1.2.2 Output parameter description

i_nbr_mb_ver The number of macro blocks in vertical direction for a picture.
i_nbr_mb_hor The number of macro blocks in horizontal direction for a picture.
i_nbr_mbs The number of macro blocks in a picture calculated as $i_nbr_mbs = i_nbr_mb_hor \cdot i_nbr_mb_ver$
i_pic_init_qp_minus26 Same definition of pic_init_qp_minus26 as defined in H.264/AVC PPS. It can be extract from H.264/AVC bitstream.

3.1.3 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 = [i_nbr_mbs,  
RTP_bitstream,  
p_mb_data_array[],  
i_pic_init_qp_minus26]
```

```
struct MB_data  
{  
    i_dct_coef_ac[][],  
    i_dct_coef_dc[],  
    i_cbp,  
    i_mb_type,  
    i_qp,  
    i_mv[][],  
    i_mv[][],  
    c_predDirection  
}
```

```
struct MB_data p_mb_data_array[i_nbr_mbs]
```

```
Output = [i_frame_num,  
i_displayorder,  
i_received_packets,  
i_lostpackets,  
i_received_bytes,  
i_frame_type,  
b_reflost0,
```

```

        b_reflost1,
        i_lostframegap,
        i_slice_qp[],      i_nbr_slice_qp,
        i_slice_size[],
        i_slice_pixel[]
    ]

```

3.1.3.1 Input parameter description

i_nbr_mbs	See section 3.1.2.2.
RTP_bitstream	The video bitstream encapsulated in RTP packets.
p_mb_data_array[]	Array of structures of type “struct MB_data” containing macro block data. The size of the array is <i>i_nbr_mbs</i> . The macro block data in the structs contain the entries described below.
i_pic_init_qp_minus26	See section 3.1.2.2.

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 H.264 syntax.
i_mb_type	Macro block type
i_qp	QP for the macro block
i_mv_x[][]	Array of horizontal motion vectors at sub macro block level. The size of the array is 3x16. The first dimension indicates two inter-prediction directions, 0 for forward prediction, 1 for backward prediction, <i>i_mv_x[2][]</i> is for normalized motion vectors that will be described in section 3.2.2.3.1. The second dimension corresponds to 16 4x4 blocks. The precision of the motion vector here is the same as that from H.264 syntax.
i_mv_y[][]	Array of vertical motion vectors at sub macro block level. The size of the array is 3x16. The first dimension indicates two inter-prediction directions, 0 for forward prediction, 1 for backward prediction. <i>i_mv_y[2][]</i> is for normalized motion vectors that will be described in section 3.2.2.3.1. The second dimension corresponds to 16 4x4 blocks.
c_predDirection	prediction direction. 0x00 = intra prediction; 0x01 = forward prediction only; 0x02 = backward prediction only; 0x03 = bi-direction prediction.

3.1.3.2 Output parameter description

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

i_displayorder	Display order of a picture in the bitstream corresponding to the picture number. It is obtained from the POC and/or frame_num syntax for the H.264/AVC bitstream encoded with main or high profile.
i_received_packets	The number of received RTP packets belonging to a picture. A new picture starts by parsing the POC and/or frame_num syntax of the H.264/AVC bitstream. See section 3.1.3.3.1 for calculation.
i_lostpackets	The number of lost RTP packets belonging to a picture. See section 3.1.3.3.1 for calculation.
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 or 5 = unknown.
b_reflost0	Boolean indicating whether a reference picture in current pictures's reference_picture_list0 is totally lost. Default/initial value is FALSE
b_reflost1	Boolean indicating whether a reference picture in current pictures's reference_picture_list1 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 POC and/or frame_num syntax of H.264.
i_slice_qp[]	Array of correctly decoded slice quantization parameter of a frame, each <i>i_slice_qp</i> is in the range of [0, 51] inclusive. The array size is <i>i_nbr_slice_qp</i> .
i_nbr_slice_qp	The number of correctly decoded slice quantization parameter of a frame. It increases by 1 if an <i>i_slice_qp</i> is obtained within a frame.
i_slice_size[]	Array of number of bytes of a slice of an <i>error free</i> Intra frame. Note that, the parameter is only calculated for each <i>error free</i> Intra frame.
i_slice_pixel[]	Array of number of pixels of a slice of an <i>error free</i> Intra frame. Note that, the parameter is only calculated for each <i>error free</i> Intra frame.

3.1.3.3 Algorithms

3.1.3.3.1 Keeping track of pictures and lost packets

A new picture *i* is recorded by detecting the change of POC and/or frame_num syntax from the H.264/AVC bitstream. The display order number value *i_displayorder* calculated from POC and/or frame_num is recorded for the picture *i*. All the RTP packets between the first packet (i.e. the packet having POC and/or frame_num syntax) of the picture *i* and the first packet (i.e. the packet having different POC and/or frame_num syntax) of the next received picture *i+1* are counted as the

RTP packets of the picture i . The number of received bytes, $i_received_bytes$, is the sum of elementary stream size in the received packets of the picture.

If one or more consecutive pictures have been lost, it can be detected by detecting the discontinuity in $i_displayorder$ of parsed pictures.

The number of lost RTP packets of the picture i is counted based on the discontinuity of sequence numbers of RTP headers. The lost RTP packets between the first received RTP packet of the picture i and the first RTP packet of the next received picture $i+1$ are counted into the lost packets of the picture i .

3.1.3.3.2 Obtain slice quantization parameter

The parameter, i_slice_qp , is the correctly decoded slice quantization parameter and is obtained per slice within a frame. Correctly decoded means that there is no packet loss occurred between the first transport packet of the slice and the transport packet which contains the bits for acquisition of $slice_qp_delta$. $slice_qp_delta$ is parsed from the slice header as defined in H.264/AVC.

$$i_slice_qp = 26 + i_pic_init_qp_minus26 + slice_qp_delta$$

3.2 Aggregation of basic parameters into internal picture level parameters

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

3.2.1 Compression module parameters

```
Input  = [i_slice_qp[],  
          i_nbr_slice_qp,  
          i_slice_size[],  
          i_slice_pixel[] ]  
  
Output = [f_frame_content_complexity[],  
          i_nbr_error_free_intra_frame,  
          i_total_slice_qp,  
          i_nbr_total_slice_qp]
```

3.2.1.1 Input parameter description

$i_slice_qp[]$	See subsection 3.1.3.2
$i_nbr_slice_qp$	See subsection 3.1.3.2
$i_slice_size[]$	See subsection 3.1.3.2
$i_slice_pixel[]$	See subsection 3.1.3.2

3.2.1.2 Output parameter description

f_frame_content_complexity[]

Array of frame content complexity of an *error free* Intra frame. Its array size is *i_nbr_error_free_intra_frame*.

i_nbr_error_free_intra_frame

The number of *error free* Intra frame of the video sequence. It increases by 1 if an *error free* frame is received.

i_total_slice_qp

The total value of *i_slice_qp* of the video sequence. It is the sum of all *i_slice_qp* of the video sequence.

i_nbr_total_slice_qp

The number of *i_slice_qp* of the video sequence. It can be calculated either by summing up *i_nbr_slice_qp* of each frame or increasing by 1 if an *i_slice_qp* is obtained.

3.2.1.3 Algorithms

3.2.1.3.1 Obtain frame content complexity

For each *error free* Intra frame, its frame content complexity (i.e., *f_frame_content_complexity*) is calculated by averaging the value of slice content complexity of all slice(s) (i.e., *f_slice_content_complexity[]*) within the *error free* Intra frame. The *f_slice_content_complexity[]* means an array of slice content complexity of the *error free* Intra frame, and its array size is the number of slice(s) of the *error free* Intra frame, i.e., *f_slice_content_complexity* is calculated per slice within the *error free* Intra frame. For each slice content complexity (i.e., *f_slice_content_complexity*) of the *error free* Intra frame, it is calculated according to its bytes per pixel (i.e., *f_slice_byte_per_pixel*).

The bytes per pixel of a slice (i.e., *f_slice_byte_per_pixel*) is calculated according to its size in bytes (i.e., *i_slice_size*) and pixels (i.e., *i_slice_pixel*) as below:

$$f_slice_byte_per_pixel_k = \frac{i_slice_size_k}{i_slice_pixel_k}$$

where *k* is from 1 to the number of slice(s) of the *error free* Intra frame, inclusive; *i_slice_size_k* means the number of bytes of the *k*-th slice; *i_slice_pixel_k* means the number of pixels of *k*-th slice; *f_slice_byte_per_pixel_k* means the bytes per pixel of *k*-th slice.

For implementation, pseudo code is provided below:

```
for( i = 0; i < number_of_slices_of_error_free_intra_frame; i++ )  
{  
    f_slice_byte_per_pixel[i] = i_slice_size[i] / i_slice_pixel[i];  
}
```

The slice content complexity of a slice (i.e. *f_slice_content_complexity*) is calculated according to its correctly decoded quantization parameter of the slice (i.e., *i_slice_qp*) and bytes per pixel of the slice (i.e., *f_slice_byte_per_pixel*) as below:

$$f_slice_content_complexity_k = a[i_slice_qp_k] * f_slice_byte_per_pixel_k + b[i_slice_qp_k]$$

where k is from 1 to the number of slice(s) of the *error free* Intra frame, inclusive; *i_slice_qp_k* means the correctly decoded slice quantization parameter of the k-th slice; *f_slice_byte_per_pixel_k* means bytes per pixel of k-th slice; *f_slice_content_complexity_k* means the slice content complexity of k-th slice; *a[]* and *b[]* are arrays holding coefficients value for each *i_slice_qp* respectively and as shown below:

For SD resolution

a[52] = { 24.78954, 24.78954, 25.23854, 25.51193, 25.74990, 25.97533, 26.19479, 26.28303, 26.49158, 26.56645, 26.53197, 26.62563, 26.69239, 26.65409, 26.79309, 26.80578, 26.84816, 27.08741, 27.25370, 27.36097, 27.56078, 27.70162, 27.85621, 28.04059, 28.17621, 28.23445, 28.41471, 28.45078, 28.54265, 28.60014, 28.62930, 28.64529, 28.74102, 28.75523, 28.76358, 28.74681, 28.77488, 28.73642, 28.79531, 28.69430, 28.72766, 28.60666, 28.49484, 28.35642, 28.07614, 27.90134, 27.57123, 27.01405, 26.65987, 26.31439, 25.52575, 25.01169};

b[52] = { 13.39250, 13.39250, 13.97091, 14.53803, 15.25528, 16.13630, 16.99497, 17.66163, 18.80068, 19.89785, 21.20091, 22.86877, 24.44105, 25.98037, 28.04957, 30.07985, 32.07935, 34.30203, 36.32256, 38.18652, 40.93258, 43.77054, 46.53546, 50.53632, 54.36178, 57.82423, 63.29899, 69.18878, 75.07466, 83.80263, 91.47496, 99.18949, 111.47580, 124.34650, 136.49900, 156.17670, 176.23080, 192.16970, 223.83720, 251.77270, 285.92790, 333.53770, 388.41820, 435.09860, 531.05070, 633.24080, 760.16820, 948.15240, 1168.53720, 1361.84570, 1759.43160, 2040.35460};

For 1280x720 resolution

a[52] = { 16.17209, 17.45819, 17.80732, 18.02041, 18.18083, 18.52479, 19.03342, 19.06581, 19.41564, 19.85189, 20.07956, 20.81183, 21.43127, 21.83287, 22.61658, 23.14807, 23.92571, 25.20184, 26.03683, 26.68701, 27.49974, 28.12203, 28.66205, 29.27020, 29.69070, 29.92960, 30.40275, 30.60385, 30.85636, 31.06785, 31.26051, 31.35589, 31.63646, 31.76881, 31.92259, 32.08798, 32.28134, 32.36179, 32.60119, 32.61653, 32.75291, 32.73418, 32.72940, 32.70158, 32.59009, 32.41000, 32.21505, 31.76353, 31.23468, 30.87401, 30.01071, 29.31316};

b[52] = { 33.81798, 33.05324, 35.11725, 36.95499, 39.10951, 41.62373, 43.87256, 45.95354, 49.32386, 51.87803, 54.92251, 58.42482, 61.62755, 64.56505, 69.19412, 73.35919, 76.10406, 78.96517, 81.95586, 84.59924, 89.05335, 93.59975, 98.31476, 105.41810, 112.34964, 118.73374, 129.00992, 140.01562, 151.12381, 167.62430, 182.02425, 196.08347, 218.72591, 241.16108, 263.35157, 295.99927, 329.06899, 355.66280, 407.64235, 452.09915, 508.72302, 585.36672, 671.43978, 741.49561, 891.18944, 1051.86892, 1246.04333, 1527.50615, 1894.63282, 2204.87735, 2879.95903, 3390.89788};

For 1920x1080 resolution (1080)

a[52] = { 15.75673, 16.17239, 17.33657, 18.09218, 18.78856, 19.85244, 20.94081, 21.42377, 25.25608, 25.36929, 25.37671, 25.59413, 25.77414, 25.89431, 26.16539, 26.37098, 26.71202, 27.45373, 27.99336, 28.43923, 29.01115, 29.49924, 29.89337, 30.32379, 30.59313, 30.74944, 31.01314, 31.10389, 31.21737, 31.28295, 31.38585, 31.36863,

31.44693, 31.40169, 31.43938, 31.39075, 31.36072, 31.33672, 31.26816, 31.16160, 31.03165, 30.80631, 30.57609, 30.36353, 30.06076, 29.62381, 29.37353, 29.05716, 28.60942, 28.52338, 28.40104, 28.52280};

```
b[52] = { 25.92973, 26.42403, 26.72231, 27.10874, 27.55908, 27.59167, 27.40409, 27.63129,
21.08740, 22.32786, 23.78112, 25.55635, 27.25511, 28.80079, 31.33600, 33.71534,
35.51380, 37.14249, 38.57997, 39.75292, 41.50986, 43.25411, 45.08496, 47.92251,
50.97660, 53.82247, 58.50549, 64.00109, 69.59487, 78.31654, 84.35147, 92.89916,
105.12040, 119.83478, 131.13182, 152.46046, 175.28796, 191.40711, 231.17849,
262.14953, 311.33306, 374.98524, 454.98602, 524.68907, 656.91124, 830.55605,
990.09180, 1196.94617, 1493.32352, 1667.34794, 1966.34090, 2099.62991};
```

For implementation, pseudo code is provided below:

```
for( i = 0; i < number_of_slices_of_error_free_intra_frame; i++ )
{
    f_slice_content_complexity[i] = a[i_slice_qp[i]] * f_slice_byte_per_pixel[i] + b[i_slice_qp[i]];
}
```

For each *error-free* Intra frame, its frame content complexity (i.e., *f_frame_content_complexity*) is calculated according to its slice content complexity of all slice(s) (i.e., *f_slice_content_complexity[]*) and its number of slice(s) as below:

$$f_frame_content_complexity = \frac{\sum_{k=1}^{Num} f_slice_content_complexity_k}{Num}$$

where Num is the number of slice(s) of the *error free* Intra frame; k is from 1 to the number of slice(s) of the *error free* Intra frame, inclusive; *f_slice_content_complexity_k* means the slice content complexity of the k-th slice.

For implementation, pseudo code is provided below:

```
float f_total_slice_content_complexity = 0.0f;
for( i = 0; i < number_of_slices_of_error_free_intra_frame; i++ )
{
    f_total_slice_content_complexity += f_slice_content_complexity[i]
}
f_frame_content_complexity = f_total_slice_content_complexity / num_of_slice_of_error_free_intra_frame
```

3.2.2 Slicing module parameters

```
Input = [i_nbr_mbs,
         f_fps,
         i_displayorder,
         i_received_packets,
         i_lostpackets,
         i_received_bytes,
         i_frame_type,
         b_reflost0,
         b_reflost1,
```

```

        i_lostframegap,
        p_mb_data_array[]

Output = [i_processed_frames,
          p_frame_parameters[]]

struct Frame_parameters
{
    i_nbr_ecmbs,
    i_frame_type,
    i_ecdist,
    b_reflost0,
    b_reflost1,
    i_refFramesIdx[]
    d_IntraRatio,
    i_totalpackets,
    i_lostpackets,
    i_totalbytes,
    i_avgbytesPerframe,
    d_avgmv,
    d_residual_var,
    b_fade_pic,
    b_scut_candidate,
    b_scut_loss,
    d_LoVA,
    struct MB_parameters *p_mb_parameters
}

struct Frame_parameters * p_frame_parameters[MAXFRAMES]

struct MB_parameters
{
    d_mvmedian,
    d_mvmedian_ref0,
    d_mvmedian_ref1,
    b_ec,
    b_mvexist,
    c_predDirection,
    b_intramode,
    d_residual_var,
    d_lova
}

```

3.2.2.1 Input parameter description

i_nbr_mbs See subsection 3.1.2.2

f_fps See subsection 3.1.1.2

p_mb_data_array[] See subsection 3.1.3.1

For all other input parameters see subsection 3.1.3.2.

3.2.2.2 Output parameter description

i_processed_frames The number of parsed frames. This variable is increased by one when a frame is parsed by the H.264/AVC parser by recording the change of i_displayorder of adjacent frames. If there is discontinuity in

i_displayorder, it means that the packets of one or several frame(s) are lost totally, thus increase *i_processed_frames* correspondingly. The details are shown in pseudo code below.

p_frame_parameters[] Array holding parameters for each parsed picture.

The following output parameters are calculated or saved from parser information for each parsed picture in *p_frame_parameters[]*:

i_nbr_ecmbs	The number of error concealed MBs in a frame.
i_frame_type	Frame type for the picture. Default/initial value is 5, i.e. unknown frame type.
i_ecdist	An error concealment distance, defined as a distance, in display order, between the to-be-concealed picture and the concealing picture (i.e., the picture used for error concealment). It is explained in subsection 3.2.2.3 how this parameter is derived. Default/initial value is a larger value, e.g., 0xFFFF.
b_reflost0	See subsection 3.1.3.2. Default/initial value is FALSE.
b_reflost1	See subsection 3.1.3.2. Default/initial value is FALSE.
i_refFramesIdx[]	The display order of the immediate forward [0] and/or backward [1] reference frame of current frame. Its dimension is 2.
d_IntraRatio	The ratio of the number of intra macro blocks to all received macro blocks, i.e., $i_nbr_intraMB / (i_nbr_mbs - i_nbr_ecmbs)$.
i_totalpackets	The total number of packets belonging to a frame. It includes the received packets and the estimate of the lost packets of the frame.
i_lostpackets	See subsection 3.1.3.2
i_totalbytes	The number of total bytes belonging to a frame. It includes the received bytes and the estimate of the bytes of lost packets.
i_avgbytesPerframe	The average number of bytes per frame calculated in a sliding window of L=10 frames. Initial value is 0.
d_avgmv	Average motion of the inter-prediction MBs in the frame. Initial value is 0.
d_residual_var	Average variance of the prediction residual signal in the frame. Initial value is 0.
b_fade_pic	Boolean indicating whether a frame have video content of gradual transition effects. The default/initial value is FALSE.
b_scut_candidate	Boolean indicating whether this frame is a scene cut candidate frame. The default/initial value is FALSE.
b_scut_loss	Boolean indicating whether a frame having packet loss is a scene cut frame. The scene cut frame can be current frame or the immediately previous reference frame that is lost completely. The initial value is FALSE.
d_LoVA	The level of visible artifacts in a frame.
p_mb_parameters[]	Pointer to an array of MB parameters. Its size is <i>i_nbr_mbs</i> .

The following output parameters are calculated or saved for each MB in *p_mb_parameters[]*.

d_mvmedian	The magnitude of the median of the available motion vectors of the macro block and its immediate adjacent macro blocks.
d_mvmedian_ref0	The magnitude of the median of the macro block's motion vectors pointed to the lost immediately forward reference frame. The default value is zero.
d_mvmedian_ref1	The magnitude of the median of the macro block's motion vectors pointed to the lost immediately backward reference frame. The default value is zero.
b_ec	Boolean indicating whether error concealment is applied to the macro block or not. Initial value is FALSE.
b_mvexist	If the <i>i_mb_type</i> is intra mode or the MB is error concealed, its value is FALSE. Otherwise, TRUE. Initial value is FALSE.
c_predDirection	See subsection 3.1.3.2. Default/initial value is 0x00.
b_intramode	Boolean indicating if this MB is encoded with intra mode.
d_residual_var	Variance (i.e. AC energy) of the prediction residual signal of the MB. Initial value is 0.
d_lova	The level of visible artifacts of the MB

3.2.2.3 Algorithms

The main components are shown in figure 3-3. First, a parsing of the sequence is made to derive frame-level parameters in structure *p_frame_parameters[]* and MB-level parameters in structure *p_mb_parameters[]*. Secondly, detect if a frame is a gradual transition frame and set *b_fade_pic* correspondingly. For each of the frames whose *i_lostpackets* > 0, detect if it is candidate scene cut frame and set *b_scut_candidate* correspondingly. If *b_scut_candidate* is true and the frame is not a gradual transition frame, further detect if the frame is a frame having scene cut artifact and set *b_scut_loss* correspondingly.

The important output parameters in this stage consist of MB-level parameters i.e. *d_mvmedian*, *d_residual_var*, and frame-level parameters i.e. *b_fade_pic*, *b_scut_candidate*, *b_scut_loss*. These parameters will be used to estimate the visible artifact level due to packet loss in each frame of the sequence. Then the frames' visible artifact levels are aggregated into a sequence-level visible artifact metric.

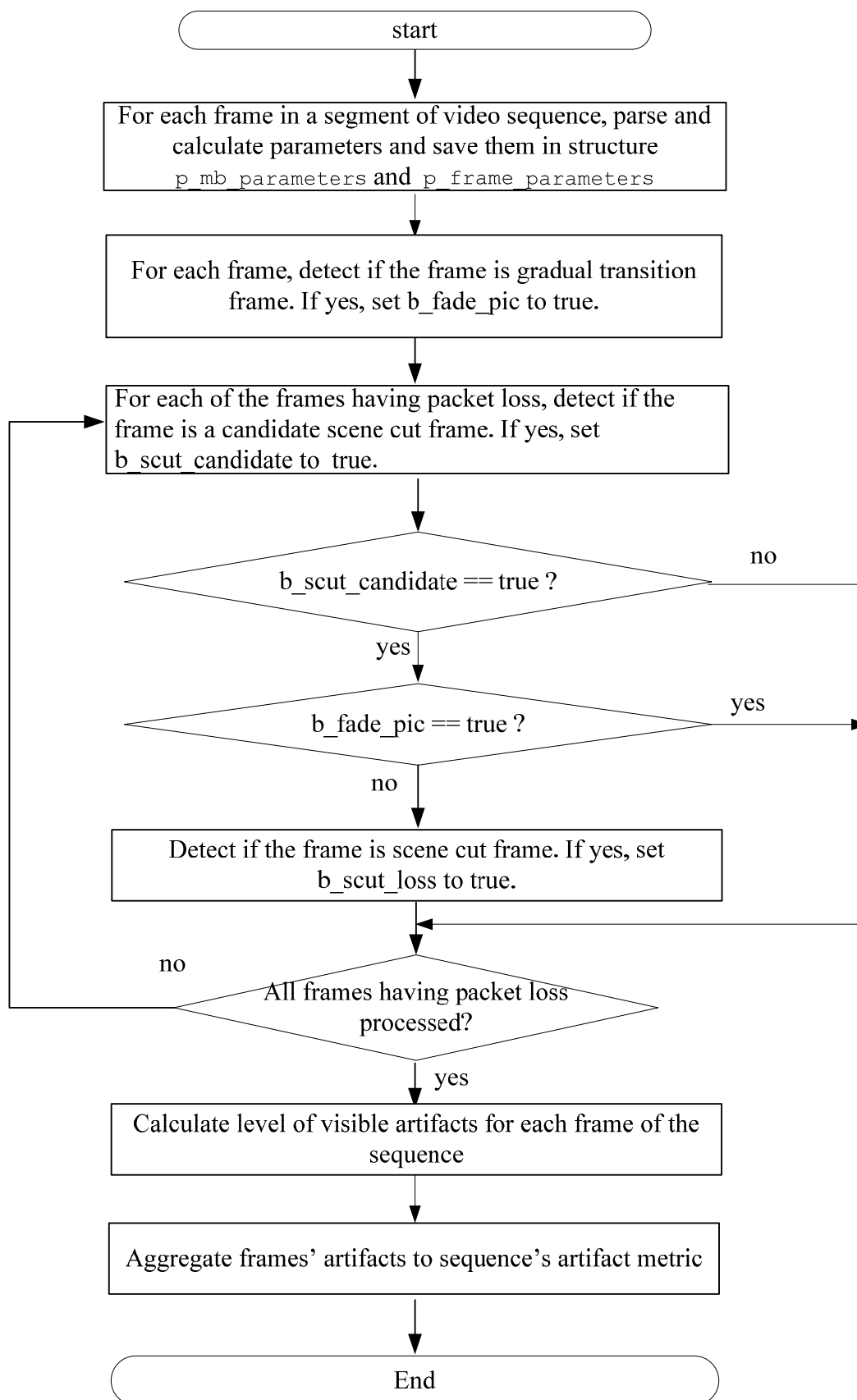


Figure 3-3 - Overview of components