International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

P.1203.1

(12/2016)

SERIES P: TELEPHONE TRANSMISSION QUALITY, TELEPHONE INSTALLATIONS, LOCAL LINE NETWORKS

Models and tools for quality assessment of streamed media

Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport – Video quality estimation module

Recommendation ITU-T P.1203.1

ITU-T P-SERIES RECOMMENDATIONS

TELEPHONE TRANSMISSION QUALITY, TELEPHONE INSTALLATIONS, LOCAL LINE NETWORKS

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
Methods for objective and subjective assessment of speech and video quality	Series	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
Methods for objective and subjective assessment of quality of services other than speech and video	Series	P.1500

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

Recommendation ITU-T P.1203.1

Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport – Video quality estimation module

Summary

Recommendation ITU-T P.1203.1 specifies the short-term video representation quality estimation modules for ITU-T P.1203 (*Pv* module). The ITU-T P.1203 series of Recommendations specifies modules for a set of model algorithms for monitoring the integral media session quality for transport control protocol (TCP) type video streaming. The models comprise modules for short-term video-quality (described in this part of the Recommendation family) and audio-quality estimation. The per-one-second outputs of these short-term modules are integrated into estimates of audio-visual quality and together with information about initial loading delay and media playout stalling events, they are further integrated into the final model output, to provide the estimate of integral quality. The respective ITU-T work item has formerly been referred to as the parametric non-intrusive assessment of TCP-based multimedia streaming quality, or P.NATS. The ITU-T P.1203.1 part of ITU-T P.1203 provides details for the modules for bitstream-based, short-term video quality estimation.

Four different modes can be used for the *Pv* module specified in this Recommendation. These modes, referred to as mode 0 to 3, use input information of differing complexity and amount and represent four model algorithms each with a different level of complexity. The *Pv* modules comprise components reflecting the effects due to video compression, up-scaling of content and the effect due to low frame rates. The four different modes use the same overall model architecture and individual coefficients and all have the same components for up-scaling and framerate. The only *Pv* module component that differs between modes is the *Pv* module component related to video compression.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T P.1203.1	2016-12-22	12	11.1002/1000/13159

Keywords

Adaptive streaming, audio, audiovisual, IPTV, mean opinion score (MOS), mobile TV, mobile video, monitoring, multimedia, progressive download, QoE, TV, video.

---

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

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/.

Table of Contents

Recommendation ITU-T P.1203.1

Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport – Video quality estimation module

1 Scope

This Recommendation describes the video module as an integral part of the ITU-T P.1203-series Recommendations. The ITU-T P.1203 Recommendations describe a set of objective parametric quality assessment modules. Combined, these modules can be used to predict the impact of audio and video media encodings as well as Internet protocol (IP) network impairments on the quality experienced by an end-user of multi-media streaming applications.

The addressed streaming techniques comprise progressive download as well as adaptive streaming, for both mobile and fixed network streaming applications over transport control protocol (TCP) or other TCP like protocols which are not affected by transmission errors.

The model described is restricted to information provided to it by an appropriate packet- or bitstream-analysis module. The overall ITU-T P.1203 model is applicable for the effects due to audio- and video-coding as well as initial buffering and re-buffering (which are both perceivable as stalling of the media) as the typical degradations associated with progressive download. As final output, the ITU-T P.1203-series models target integral audiovisual media quality scores.

This Recommendation describes four different quality modules, one for each mode of [ITU-T P.1203], that is, modes 0, 1, 2 and 3. The video quality module predicts mean opinion scores (MOS) on a 5-point ACR scale (see [ITU-T P.910]) as a per-one-second MOS score. The underlying sliding window is described in [ITU-T P.1203]. If used stand-alone, the video module can provide estimates of short-term video quality at per-one-second intervals.

This model cannot provide a comprehensive evaluation of video transmission quality as perceived by an individual end user because its scores reflect the impairments due to video scaling, framerate and coding only. Furthermore, the scores predicted by a parametric model necessarily reflect an average perceptual impairment.

Note also that the model was developed and validated for one specific encoder and decoder implementation. If a different encoder and decoder pair is used in a monitoring situation the scores may not reflect that.

Effects such as flicker due to low source bitrate or other, not coding or transmission caused impairments related to the payload are not reflected in the scores computed by this model. Moreover, the scores predicted by a parametric model (especially in case of no access to payload or pixel information) necessarily reflect a somewhat simplified representation of the perceptual impairment of the considered stream.

However, the model still enables estimation of some coding quality related information and thus valid and in most cases accurate predictions, presuming that it is applied in an appropriate manner, following this Recommendation.

Table 1-1 shows application areas, test factors and coding technologies where ITU-T P.1203.1 for adaptive streaming and progressive download has been verified and is known to produce reliable results.

Table 1-1 – Application areas, test factors and coding technologies where ITU-T P.1203.1 for adaptive streaming and progressive download has been verified and is known to produce reliable results

Applications for which the model is intended	
In-service monitoring of audiovisual, TCP-based video and audio. Both so called over-the-top (OTT) services (for example YouTube) and operator managed video services (over TCP), using the protocols HTTP/TCP/IP and RTMP/TCP/IP. Note that this model is agnostic to the type of container format (Flash (FLV), MP4, WebM and 3GP).	
Performance and quality assessment of live networks (including codecs) considering the effect due to encoding bitrate	
Video test factors for which the model has been validated	
Video content	Movie trailers, sports videos, documentaries, freely available HD content, time lapse videos, etc.
Input video length	20 seconds. The video model produces a per-second score considering a 20-second sliding window
Bitstream container	Elementary stream contained in transport stream (TS) segments
Encoder/decoder implementation	The model has been trained using the following video encoder/decoder: – ITU-T H.264/MPEG-4 AVC High profile: x264 (ffmpeg) – A common framework was developed based on the above codec, all the test data was generated using the common framework. It is assumed that the model can be used for estimating quality when other encoder implementations for the given codec have been used. However, model performance cannot be guaranteed in this case.
Slice size	1 slice per video frame
Scene-cut detection	Off
Preset	Medium
Video resolution/bitrate	240p: 75-150 kbit/s 360p: 220-450 kbit/s 480p: 375-750 kbit/s 720p: 1 050-2 100 kbit/s 1080p: 1 875-12 500 kbit/s Note that aspect ratio of 16:9 is maintained for all quality levels.
Group of pictures (GOP)	1 second length, IBBBP only
TS segment length	1-9 seconds NOTE – The segment length determines how often the quality can be adapted.

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 G.107] Recommendation ITU-T G.107 (2015), *The E-model: a computational model for use in transmission planning*.

[ITU-T H.264] Recommendation ITU-T H.264 (2016), *Advanced video coding for generic audiovisual services*.

[ITU-T P.800.1] Recommendation ITU-T P.800.1 (2006), *Mean opinion score (MOS) terminology*.

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

[ITU-T P.911] Recommendation ITU-T P.911 (1998), *Subjective audiovisual quality assessment methods for multimedia applications*.

[ITU-T P.1201.1] Recommendation ITU-T P.1201.1 (2012), *Parametric non-intrusive assessment of audiovisual media streaming quality – Lower resolution application area*.

[ITU-T P.1201.2] Recommendation ITU-T P.1201.2 (2012), *Parametric non-intrusive assessment of audiovisual media streaming quality – Higher resolution application area*.

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

[ITU-T P.1202.1] Recommendation ITU-T P.1202.1 (2012), *Parametric non-intrusive bitstream assessment of video media streaming quality – Lower resolution application area*.

[ITU-T P.1203] Recommendation ITU-T P.1203 (2016), *Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport.*

[ITU-T P.1203.2] Recommendation ITU-T P.1203.2 (2016), *Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport – Audio quality estimation module.*

[ITU-T P.1203.3] Recommendation ITU-T P.1203.3 (2016), *Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport – Quality integration module.*

[ITU-T P.1401] Recommendation ITU-T P.1401 (2012), *Methods, metrics and procedures for statistical evaluation, qualification and comparison of objective quality prediction models*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 mean opinion score (MOS): [ITU-T P.800.1].

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 model, model algorithm: An algorithm with the purpose of estimating the subjective (perceived) quality of a media sequence.

3.2.2 sequence: A short decoded audio, video or audiovisual portion of a stream, typically shorter than 30 seconds.

3.2.3 bitstream: The part of an IP-based transmission where the actual audiovisual, video, or audio content is available in encoded and packetized form.

3.2.4 compression artefacts: Artefacts introduced due to lossy compression of the encoding process.

3.2.5 media adaptation: Refers to events where the player switches video playback between a known set of media quality levels while adapting to network conditions.

3.2.6 initial buffering: Refers to the time in seconds between the initiation of video playback by the user and the actual start of the playback.

3.2.7 rebuffering or stalling artefacts: Artefacts coming from rebuffering events at the client side, which could be a result of video data arriving late. Usually, stalling events are indicated to the viewer, e.g., in the form of a spinning wheel.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AAC Advanced Audio Coding

ARQ Automatic Repeat Request

DASH Dynamic Adaptive Streaming over HTTP

FEC Forward Error Correction

GOP Group Of Pictures

HD High Definition (television)

IP Internet Protocol

MOS Mean Opinion Score

MPEG Moving Pictures Expert Group

NTSC National Television Systems Committee

OTT Over The Top

PCC Pearson Correlation Coefficient

RBSP Raw Byte Sequence Payload

RMSE Root Mean Square Error

SD Standard Definition

TS Transport Stream

TCP Transport Control Protocol

UDP User Datagram Protocol

5 Conventions

None.

6 *Pv* module in ITU-T P.1203 context

The overall model structure is shown in Figure 6-1, highlighting the position of the *Pv* module. More details on the general structure can be found in [ITU-T P.1203].

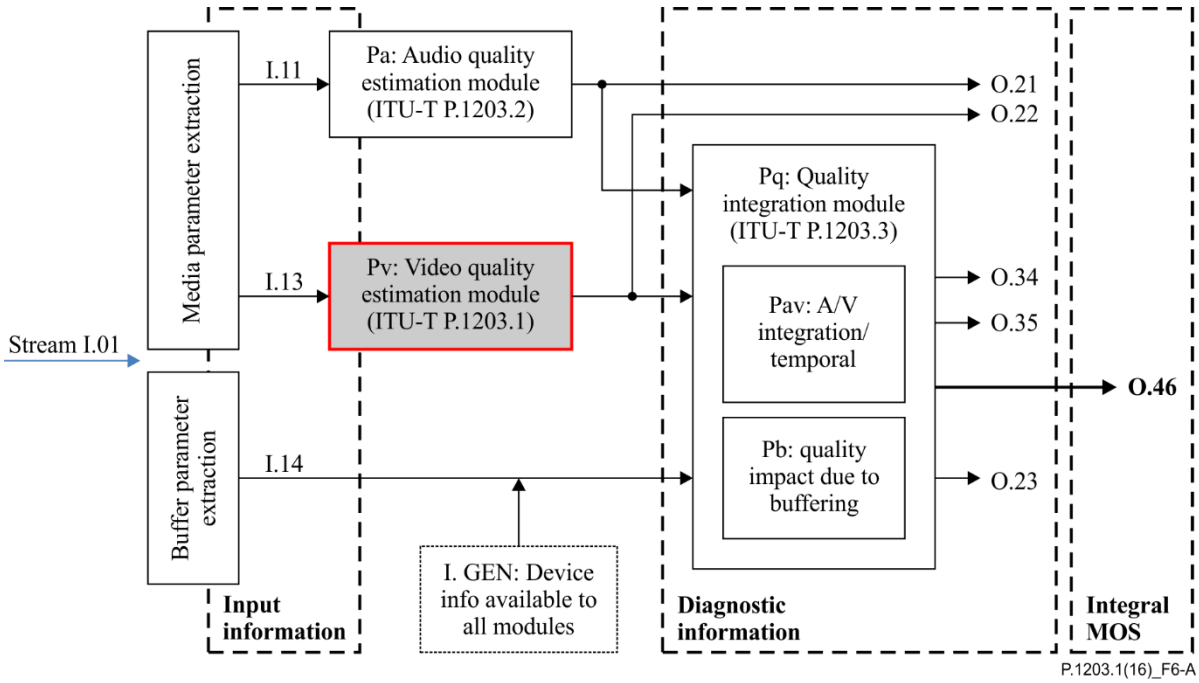

Figure 6-1 – *Pv* module in context of building blocks of the ITU-T P.1203 model

6.1 Pv module modes

The modes of operation for ITU-T P.1203.1 are defined in Table 6-1. Detailed information on exactly which inputs are available for each mode is provided in Table 7-1. Individual models are specified per mode and are described in Annexes A, B, C, D.

Table 6-1 – ITU-T P.1203.1 modes of operation

Mode	Encryption	Input	Complexity
0	Encrypted media payload and media frame headers	Meta-data	Low
1	Encrypted media payload	Meta-data and frame header information	Low
2	No encryption	Meta-data and up-to 2% of the media stream	Medium
3	No encryption	Meta-data and any information from the video stream	Unlimited

7 Model input

The model receives media information and prior knowledge about the media stream. In various modes of operation, inputs may be extracted or estimated in different ways, which are outside the scope of this Recommendation but which may be added in future annexes. The core model receives the following input signals regardless of the mode of operation:

I.13: video coding information, as specified in [ITU-T P.1203.2].

Note that fault correction techniques, such as automatic repeat request (ARQ) and forward error correction (FEC) used for user datagram protocol (UDP) based streaming are not applicable in this case, where the streaming is TCP based. In TCP-based transport all retransmissions and packet loss information is typically handled transparently by the transport layer and while it can be available to the models described in this Recommendation it is not needed. Any packet loss or packet retransmissions are conveyed to the models described in this Recommendation as latency.

7.1 I.13 input specification

Table 7-1 – Description of I.13

ID	Description	Values	Frequency	Modes
I.13				
12	Target Video chunk bit-rate	Bit-rate in kbit/s	Per .TSmedia segment video chunk	All
13	Video chunk frame-rate	Frame rate in frames per second	Per video chunk.TSmedia segment	All
14	Video chunk Segment duration	Duration in seconds	Per video chunkPer.TSmedia segment	All
15	Video encoding resolution per TS segments	Number of pixels (WxH) in transmitted video	Per video chunk/frame.TSmedia segment	All
16	Video codec and profile	One of: H264-baseline, H264-high, H264-main	Per video chunk/frame.TSmedia segment	All
17	Video frame number	Integer, starting at 1, denoting frame sequence number in encoding order.	Per video frame	1,2,3
18	Video frame duration	Duration of the frame in seconds	Per video frame	1,2,3
198	Frame presentation timestamp	The frame presentation timestamp	Per video frame	1,2,3
20	Frame decoding timestamp	The frame decoding timestamp	Per video frame	1,2,3
21	Video frame size	The size of the encoded video frame in bytes	Per video frame	1,2,3
22	Type of each picture	"I" or "Non-I" for mode 1	Per video frame	1
23	Video bitstream	Encoded video bytes for the frame.	Per video frame	2,3

8 Model algorithm and output

The video module defined in this Recommendation has one output, O.22. It provides output values on the 5-point ACR scale ("MOS") per output sampling interval.

There is a core model algorithm, which is described hereafter. It comprises four different modes, which all are based on the same basic function. For all four modes, the degradation due to spatial up-scaling of content that is streamed at lower resolution as well as the handling of temporal/ jerkiness-related degradations due to low video framerates are the same. Only a third degradation term addressing video compression artefacts, referred to as *Dq* in Clause 8.1.1, is mode-dependent.

The core model as well as the terms that are mode-independent are described in this clause. The model component addressing coding-related degradations, the "quantization-related component" that depends on the respective mode, is given in Annexes A (mode 0) through D (mode 3).

The following parameters are used in the description of the model:

- *quant*, $quant \in [0,1]$: a parameter capturing the quantization degradation; its computation depends on the variant of the quantization degradation module.
- *scaleFactor*, $scaleFactor \in [0,1]$: a parameter capturing the upscaling degradation.
- *framerate*: the video frame rate in frames per second.
- *disRes*: the video display resolution in number of pixels. For instance, high definition (HD) 1080p screens have a video display resolution of $disRes = 1920 \times 1080 = 2073600\ pixels$.
- *codRes*: the video encoding resolution in pixels. The video resolution used to encode the video, for instance, $codRes = 854 \times 480 = 409920\ pixels$.
- *displaySize:* the video display diagonal length, in cm or inches.
- *deviceType:* a parameter describing how the device is used; if the video display is handheld/mobile or stationary, such as a PC or TV. This parameter is preferably set based on knowledge about the device but can also be estimated from the *displaySize* parameter if it is unknown. In the case of unknown device, *deviceType* should be set to "handheld" if *displaySize* is smaller than 33 cm (13"). If in the same scenario it is equal to or larger than 33 cm, *deviceType* is "TV".

8.1 Core model

Figure 8-1 shows the *Pv* module in the context of building blocks of the ITU-T P.1203 model.

8.1.1 Quantization degradation

$$Dq = 100 - RfromMOS(\widehat{MOSq}) \quad (1)$$

where $RfromMOS$ is described in Annex E (as in [ITU-T G.107]), and

$$Dq = \max(\min(Dq, 100), 0)$$

and where

$$\widehat{MOSq} = q_1 + q_2 \cdot \exp(q_3 \cdot quant) \quad (2)$$

and

$$\widehat{MOSq} = \max(\min(\widehat{MOSq}, 5), 1)$$

with q_1 = 4.66, q_2 = −0.07 and q_3 = 4.06.

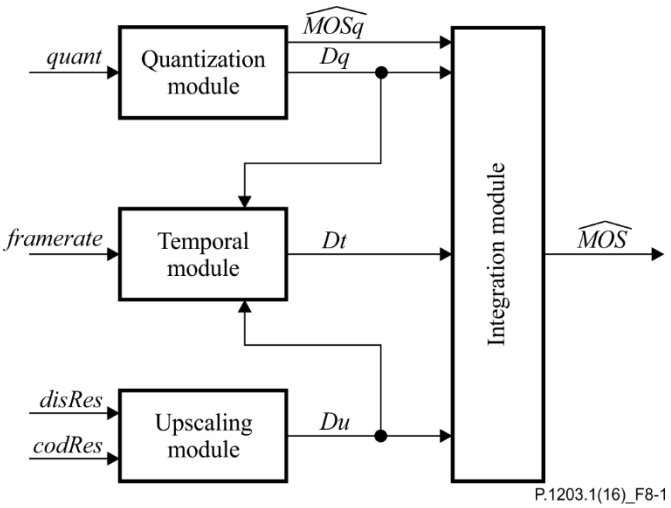

Figure 8-1 – *Pv* module in the context of building blocks of the ITU-T P.1203 model

8.1.2 Upscaling degradation

$$Du = u_1 \cdot \log_{10}(u_2 \cdot (\text{scaleFactor} - 1) + 1)) \tag{3}$$

with

$$Du = \max(\min(Du, 100), 0)$$

and

$$scaleFactor = \max\left(\frac{disRes}{codRes}, 1\right) \tag{4}$$

with u_1 = 72.61 and u_2 = 0.32.

8.1.3 Temporal degradation

$$D_t = \begin{cases} D_{t1} - D_{t2} - D_{t3}, & framerate < 24 \\ 0, & framerate \geq 24 \end{cases} \tag{5}$$

Pure temporal degradation

$$D_{t1} = \frac{100 \cdot (t_1 - t_2 \cdot framerate)}{t_3 + framerate} \tag{6}$$

Compensation variable 1, relative to coding impact

$$D_{t2} = \frac{D_q \cdot (t_1 - t_2 \cdot framerate)}{t_3 + framerate} \tag{7}$$

Compensation variable 2, relative to spatial scaling impact

$$D_{t3} = \frac{D_u \cdot (t_1 - t_2 \cdot framerate)}{t_3 + framerate} \tag{8}$$

$$D_t = \max(\min(D_t, 100), 0)$$

where t_1 = 30.98, t_2 = 1.29 and t_3 = 64.65.

8.1.4 Integration

Degradation (*D*) is calculated based on the quantization (*Dq*), upscaling (*Du*) and temporal (*Dt*) degradations as follows:

$$D = \max(\min(Dq + Du + Dt, 100), 0) \tag{9}$$

Maximum quality according to the pure upscaling and temporal degradations is calculated as follows:

$$\widehat{Qmax} = \begin{cases} 100 - Du - D_{t1}, & framerate < 24 \\ 100 - Du, & framerate \geq 24 \end{cases} \quad (10)$$

Then, quality (Q) is calculated as follows:

$$\hat{Q} = 100 - D = \begin{cases} 100 - \max\left(min\left(\left(100 - \widehat{Qmax}\right) + Dq - Dt2 - Dt3{,}100\right), 0\right), framerate < 24 \\ 100 - \max\left(min\left(\left(100 - \widehat{Qmax}\right) + Dq, 100\right), 0\right), framerate \geq 24 \end{cases} \quad (11)$$

Finally,

$$\widehat{MOS} = \begin{cases} \widehat{MOSq}, & if\ Du = 0\ and\ Dt = 0 \\ MOSfromR(\hat{Q}), & \text{otherwise} \end{cases} \quad (12)$$

where $\widehat{MOS}$ and $\hat{Q}$ are the estimated video encoding qualities on two different scales: $\widehat{MOS} \in [1,5]$ and $\hat{Q} \in [0,100]$.

$MOSfromR$ is described in Annex E.

D is the overall degradation due to video representation (spatial and temporal sampling, video compression).

If deviceType is "handheld", the following adjustment should be made on the final score:

$$\widehat{MOSq}_{handheld} = htv1 + htv2 \cdot \widehat{MOSq} + htv3 \cdot \widehat{MOSq}^2 + htv4 \cdot \widehat{MOSq}^3$$
$$\widehat{MOSq} = \max(\min(\widehat{MOSq}_{handheld}, 5), 1) \quad (13)$$

The four coefficients in the tv/handheld-adjustment are:

Coefficient	Value
*htv*1	−0.60293
*htv*2	2.12382
*htv*3	−0.36936
*htv*4	0.03409

It is noted that higher *D*, *Dq*, *Du* and *Dt* values mean higher degradations and therefore lower quality. Higher $\hat{Q}$and $\widehat{MOS}$ values mean higher quality.

In the following Annexes A through D, different variants of the quant parameter are described, one for each of the four modes.

The initial output delivered addresses the case of presentation on a PC screen of 1920x1080 pixel resolution. To estimate the O.22 output delivered by the video quality estimation module Pv for the case of viewing on a mobile device, the following adjustment of the output score is proposed.

Annex A

Mode 0 *Pv* module: *quant* description

(This annex forms an integral part of this Recommendation.)

In the case of mode 0, *quant* is calculated as described in Equation (A.2):

$$quant = a_1 + a_2 \cdot \ln(a_3 + \ln(br) + \ln(br \cdot bpp + a_4)) \quad \text{(A.1)}$$

and *bpp* is the number of kbits per pixel

$$bpp = \frac{br}{codRes \cdot fr} \quad \text{(A.2)}$$

where:

fr : framerate in frames per second

codRes : coding resolution in number of pixels of the frame

br : video stream bitrate in kbit/s

In case of TS-segments, the bitrate can be estimated using information about the chunk size, setting

$$br = brChunkSize \quad \text{(A.3)}$$

brChunkSize : estimated encoding video bitrate in kbit/s, computed from the chunk size as follows:

$$brChunkSize = \frac{chunkSize \cdot 8 - audioSize - tsHeader - pesHeader}{videoDur \cdot 1000} \quad \text{(A.4)}$$

where:

chunkSize : chunk (audio + video) size in Bytes

videoDur : duration of the video chunk in seconds

and where

$$tsHeader = 4 \cdot 8 \cdot \frac{chunkSize}{188} \quad \text{(A.5)}$$

and

$$pesHeader = 17 \cdot 8 \cdot (numVideoFrames + numAudioFrames) \quad \text{(A.6)}$$

where:

numVideoFrames : number of video frames in the chunk

numAudioFrames: number of video and audio frames in the chunk.

numVideoFrames and *numAudioFrames* are computed as follows:

$$numVideoFrames = int(ceil(videoDur \cdot framerate)) \quad \text{(A.7)}$$

and

$$numAudioFrames = int(ceil\left(audioDur \cdot \frac{audioSampleRate}{samplesPerFrame}\right)) \quad \text{(A.8)}$$

where:

audioSampleRate: audio sample rate in Hz, e.g., 48000

samplesPerFrame: number of samples per audio frame, e.g., 1024 for AAC

audioDur : audio duration within the chunk in seconds

Further:

$$audioSize = audioBrTarget \cdot audioDur \cdot 1000 \quad \text{(A.9)}$$

where:

audioBrTarget: targeted encoding audio bitrate in kbit/s

If other means of determining the video stream bitrate are available, these may be used instead of the above calculation approach, which uses the chunk sizes.

Table A.1 shows coefficient sets for *quant* parameter calculation for mode 0.

Table A.1 – Coefficient sets for *quant* parameter calculation for Mode 0

Coefficient	Value
a_1	11.99835
a_2	−2.99992
a_3	41.24751
a_4	0.13183

A.1 Performance analysis

See [ITU-T P.1203] for details on the performance of the complete model with the mode 0 *Pv* module.

Annex B

Mode 1 *Pv* module: *Dq* description

(This annex forms an integral part of this Recommendation.)

For mode 1, the *Dq* estimation for mode 0, that is, the estimation of the degradation due to coding, is amended by a frame-information-dependent term that addresses the complexity of the video content. This way, the prediction is improved over mode 0, by considering how contents with different degrees of complexity lead to a modified encoding effectiveness.

As shown in the equations below, the modification is carried out outside of the mode 0 *Dq*-calculation and not made part of the *quant* parameter. As a consequence, Annex B describes the calculation of the complete *Dq* for this case, instead of only the calculation of the *quant* parameter. This is in contrast to the modes 0, 2 and 3, where all effects covered by the respective module instantiation are captured by *quant* directly.

B.1 *quant* calculation

In the case of mode 1, *quant* is calculated as described in Equation (B.4) for mode 2, but with different coefficient values than in the case of mode 0 (see Annex A):

$$quant = a_1 + a_2 \cdot \ln(a_3 + \ln(br) + \ln(br \cdot bpp)) \tag{B.1}$$

and *bpp* is the number of kbits per pixel

$$bpp = \frac{brFrameSize}{codRes \cdot fr} \tag{B.2}$$

where:

fr : framerate in frames per second

br FrameSize

Here, *bpp* is calculated based on the parameter *brFrameSize*, the video bitrate in kbit/s computed from the video frame sizes.

brFramesize is computed as:

$$brFramesize = \frac{\sum_{index=1}^{numVideoFrames} frameSizes(index) \cdot 8}{frameDuration \cdot numVideoFrames \cdot 1000} \tag{B.3}$$

Definitions:

- *frameSizes*: Vector containing the frame size in bytes for each frame in the video chunk.
- *frameDuration*: Video frame duration in seconds, which is the same value for all frames in the video chunk.
- *numVideoFrames*: Number of video frames in the chunk.

Table B.1 shows coefficient sets for *quant* parameter calculation for mode 1.

Table B.1 – Coefficient sets for *quant* parameter calculation for mode 1

Coefficient	Value
a_1	5.00012
a_2	−1.19631
a_3	41.35850

B.2 *Dq* calculation for mode 1

Dq is calculated in the same manner as indicated in clause 8.1.1. For mode 1, however, there is an external correction factor not handled in the *quant* parameter, as outlined above, to include the effect of content-complexity.

$$Dq = 100 - RfromMOS(\widehat{MOSq}) \tag{B.4}$$

(compare with clause 8.1.1 Equation (1)), where $RfromMOS$ is described in Annex E (same as in [ITU-T G.107]) and

$$Dq = \max(\min(Dq, 100), 0)$$

and where

$$\widehat{MOSq}_1 = q_1 + q_2 \cdot \exp(q_3 \cdot quant) \tag{B.5}$$

with the coefficients shown in Table B.2:

Table B.2 – Coefficients and values

Coefficient	Value
q_1	4.66
q_2	−0.07
q_3	4.06

In addition, we set

$$iFrameRatio = \frac{\bar{I}}{\bar{I}_n} \tag{B.6}$$

with

$\bar{I}$: Average size of I-frames in chunk

$\bar{I}_n$: Average size of non-I-frames in chunk

$$\widehat{MOSq} = \widehat{MOSq}_1 + \text{sigmoid}(c0, c1, c2, c3, iFrameRatio) \tag{B.7}$$

where $\text{sigmoid}(k0, k1, k2, iFrameRatio)$ is defined with a number of input parameters as follows:

$$scalex = \frac{10}{k2 - k1} \tag{B.8}$$

$$midx = \frac{k1 + k2}{2} \tag{B.9}$$

$$\text{sigmoid}(k0, k1, k2, iFrameRatio) = k0 - \frac{k0}{1 + e^{-scalex \cdot (iFrameRatio - midx)}} \tag{B.10}$$

The modelling coefficients in the sigmoid-function are set as shown in Table B.3:

Table B.3 – Settings of the modelling coefficients in the sigmoid-function

Coefficient	Value
k_0	−0.91562479
k_1	−3.28579526
k_2	20.4098663

The final score is calculated using the bounding function:

$$\widehat{MOSq} = \max(\min(\widehat{MOSq}, 5), 1) \tag{B.11}$$

If deviceType is "handheld", the following adjustment should be done on the final score:

$$\widehat{MOSq}_{handheld} = htv1 + htv2 \cdot \widehat{MOSq} + htv3 \cdot \widehat{MOSq}^2 + htv4 \cdot \widehat{MOSq}^3$$

$$\widehat{MOSq} = \max(\min(\widehat{MOSq}_{handheld}, 5), 1) \quad \text{(B.12)}$$

The four coefficients in the tv/handheld-adjustment are shown in Table B.4:

Table B.4 - Four coefficients in the tv/handheld-adjustment

Coefficient	Value
htv1	−0.60293
htv2	2.12382
htv3	−0.36936
htv4	0.03409

B.3 Performance analysis

See [ITU-T P.1203] for details on the performance of the complete model with the mode 1 *Pv* module.

Annex C

Mode 2 *Pv* module: *quant* description

(This annex forms an integral part of this Recommendation.)

C.1 Consideration on parameter extraction

In the case of mode 2, only 2% of bitstream data is allowed to be used. This has some implications for the way in which bitstream information is being parsed. Because the video module is run separately and the audio module used for all modes corresponds to the light-weight mode 0, the byte limit is enforced on video streams only. These considerations are described in the following:

Payload not considered in the video byte count

- Container metadata (e.g., TS headers, MP4 non-MDAT boxes, etc.)
- [ITU-T H.264] Annex B start codes
- Sequence parameter sets and picture parameter sets
- The `nal_unit_type`, the `forbidden_zero_bit` and the `nal_ref_idc` indicator, i.e., the complete NAL unit header

In other words, the bytes counting towards the percentage limit correspond to the raw byte sequence payload (RBSP) as defined in [ITU-T H.264], including emulation prevention bytes in the NAL payload.

C.1.1 Implications of 2%-limitation for frame-specific information

It is noted that it may happen that some frames cannot be decoded, e.g., due to corrupted frames, bugs in the decoding software or if 2% of the frame is not enough information in order to extract information from a single frame. To compensate for such cases, the model includes a so called "frame compensation", which will allow the feature *quant* to be estimated more reliably. The frame compensation shall be run directly after bitstream parsing and before the estimation of the feature *quant*.

To do this compensation, for every output sampling interval, first all frames within a chunk will be analysed whether there are bitstream statistics available or not, using the targeted 2% of bytes per frame. During this analysis, a vector *statsVec* of length *N*, where *N* is the number of frames in the chunk, is created consisting of a list of *N* tuples, where each tuple contains:

- a boolean value and
- the frame type

per position/frame within this chunk. The *statsVec*'s frame types component can adopt the following values:

- "I", "P", "B", used when there are bitstream statistics available
- "I", "Non-I" where when those statistics are not present.

Figure C.1 shows the overall pipeline of the compensation procedure:

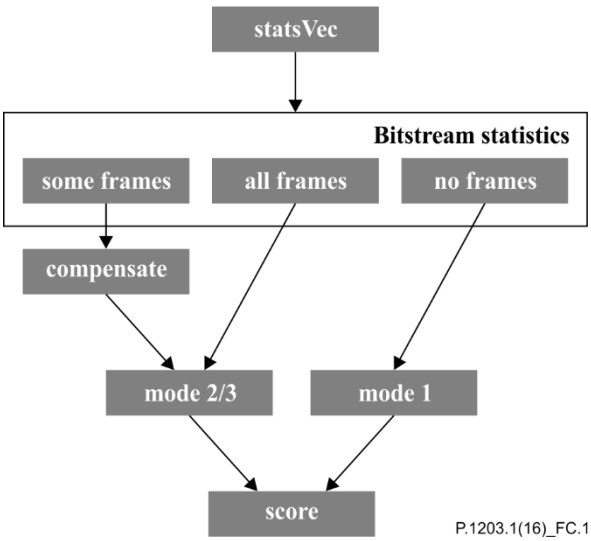

Figure C.1 – Relation between frame information and workaround by going back to mode 1 in the case of "no frames have stats" for a given chunk / model window

The previously created *statsVec* will now be analysed and classified dependent on the number of bitstream statistics available in the current chunk. There are three scenarios:

1 No frame has bitstream statistics

2 All frames have bitstream statistics

3 Some frames have bitstream statistics

No frames have bitstream statistics

In this case the model uses the algorithm specified in mode 1 for the calculation of the feature *quant*.

All frames have bitstream statistics

Since all frames can be decoded, there is no need for compensation. Therefore the algorithm for the calculation of *quant* in mode 2/3 is applied directly.

Some frames have bitstream statistics

Since some frames have bitstream statistics it is possible to compensate for frames where no bitstream statistics are available in order to avoid the drawbacks of falling back to the mode 1 algorithm. This compensation is done in four steps:

1 Iterate over all the frames in the chunk and generate *statsVec* (as described above).

2 Iterate over *statsVec* starting at position *pos* = 0 where 0 is the index of the first frame in the chunk.

3 If the frame at *pos* = 0 has statistics:

   a) Continue iterating *pos* until a frame without bitstream statistics is reached, store *statsPosLastFound* = *pos* −1.

   b) Continue iterating *pos* until a frame with bitstream statistics is reached. The area between *pos* and *statsPosLastFound* + 1 is now called a "hole", i.e., a contiguous region of frames without bitstream statistics.

   c) For every frame within the hole at position *pos* with type *type*, call the function *findNearest(pos, type)* (as described below) in order to fill the bitstream statistics.

4 If the frame at *pos* = 0 has no bitstream statistics:

   a) Set statsPosLastFound to −1

b) Continue with step 3.b.

findNearest(pos, type)

This function searches for the nearest frame (starting at position *pos*) having bitstream statistics, which has the type *type*. If no frame matching these criteria can be found, the search is performed again, this time allowing any frame type to be matched.

C.2 *quant* calculation

The quant parameter is calculated using the following formula:

$$quant = \frac{\overline{QP_{PB}}}{51} \tag{C.1}$$

Where $\overline{QP_{PB}}$ is the average over all values in the vector QP_{PB}, where QP_{PB} is computed as described in *pseudocode1* below.

Definitions:

- $initialQP$: vector containing the initial quantization parameter (QP, with $QP \in [0, 51]$) value for each frame in the video chunk;
- $twoPercentQP$, a vector containing, for each frame in the video chunk, the average of the QP values for each complete macroblock (MB) contained in the first 2% (in bytes) of the current frame's payload. If the frame contains black borders on the top and bottom of the frame (so-called "letterboxing"), the QP values of the MBs contained in these black borders are ignored;
- $frameTypes$, a vector containing the frame type for each video frame in the chunk.

Here, video frames are parsed in their order of arrival and are therefore captured in decoding order, not in presentation/display order.

Pseudocode1: computation of QP_{PB}

```
Note: Array indices start at 0.

frameTypes ← [I,P,B,…]
QPPB ← []
p ← 2
index ← 0

FOR frameType IN frameTypes:
  IF (frameType = P) OR (frameType = B):
    IF (twoPercentQP[index] = 0) OR
    (twoPercentQP[index] could not be computed) OR
    (p% for the frame contains only MBs in black border region):
      QPPB[index] ← initialQP[index]
    ELSE:
      QPPB[index] ← twoPercentQP[index]
    index ← index + 1
```

C.3 Performance analysis

See [ITU-T P.1203] for details on the performance of the complete model with the mode 2 *Pv* module.

Annex D

Mode 3 *Pv* module: *quant* description

(This annex forms an integral part of this Recommendation.)

$$quant = \frac{\overline{QP_{PB}}}{51} \tag{D.1}$$

where $\overline{QP_{PB}}$ is the average over all values in the vector QP_{PB}. For mode 3, QP_{PB} is computed as described in *pseudocode2* below.

Definitions:

- $numMBdec$: Vector containing the number of decoded MBs in the current frame for each video frame in the chunk.
- $numMBskip$: Vector containing the number of skipped MBs in the current frame for each video frame in the chunk.
- $averageQP$: Vector containing, for each video frame in the chunk, the average over the current frame of the QP values per MB, ignoring MBs in the black border areas, if the frame contains black borders.
- $frameTypes$: Vector containing the frame types for each video frame in the chunk.

Video frames are parsed from the bitstream and are therefore captured in decoding order.

Pseudocode2: computation of QP_{PB}

```
Note: Array indices start at 0.

frameTypes ← [I,P,B,…]
QPP ← []
QPB ← []
QPPB ← []
index ← 0

FOR frameType IN frameTypes:

  CASE frameType = P:

    IF  (length(QPP) = 0) OR
        (numMBskip[index] / numMBdec[index] < 0.99):
          QPtmp ← averageQP[index]
    QPP.append(QPtmp)

  CASE frameType = B:
    QPB.append(averageQP[index])

  CASE (frameType = I) AND (length(QPP) > 0):

    IF length(QPP) > 1:
      QPP[length(QPP)-1] ← QPP[length(QPP)-2]
    ELSE:
      QPP ← []

  index ← index + 1
QPPB ← QPP
QPPB.append(QPB)
```

D.1 Performance analysis

See [ITU-T P.1203] for details on the performance of the complete model with the mode 3 *Pv* module.

Annex E

RfromMOS and *RfromMOS* functions

(This annex forms an integral part of this Recommendation.)

$MOSfromR$ is expressed as follows:

$$MOSfromR: \mathbb{R} \longmapsto \mathbb{R}$$

$$Q \longmapsto MOS := MOSfromR(Q)$$

if Q > 0 and Q < 100:

MOS = MOS_MIN + (MOS_MAX-MOS_MIN)*Q/100 + Q*(Q-60)*(100-Q)*0.000007

else if Q >= 100:

MOS = MOS_MAX

else if Q <=0:

MOS = MOS_MIN

where MOS_MAX = 4.9 and MOS_MIN = 1.05

$RfromMOS$ is expressed as follows:

$$RfromMOS: \mathbb{R} \longmapsto \mathbb{R}$$

$$MOS \longmapsto Q := RfromMOS(MOS)$$

if MOS > 4.5:

MOS = 4.5

if MOS > 2.7505:

$$h = \frac{1}{3} \cdot \left(\pi - \arctan\left(\frac{15 \cdot \sqrt{-903522 + 1113960 \cdot \mathrm{MOS} - 202500 \cdot \mathrm{MOS}^2}}{6750 \cdot MOS - 18566} \right) \right)$$

else:

$$h = \frac{1}{3} \cdot \arctan\left(\frac{15 \cdot \sqrt{-903522 + 1113960 \cdot \mathrm{MOS} - 202500 \cdot \mathrm{MOS}^2}}{18566 - 6750 \cdot \mathrm{MOS}} \right)$$

with:

$$Q = \frac{20 \cdot (8 - \sqrt{226} \cdot cos\left(h + \frac{\pi}{3}\right))}{3}$$

Annex F

Byte-counting algorithm for Mode 2

(This annex forms an integral part of this Recommendation.)

Byte-counting algorithm

Below is pseudocode that illustrates the expected behaviour of a conformant byte-counting implementation:

```
pv = PVModule(metaData)
perFrameBytesConsumed = []
bytesConsumed = 0
totalBytes = 0
for h264File in h264FileList:
    byteStream = open(h264File)
    while not eof:
        consume start code at start of byteStream
        nalUnitType = getNalUnitType(byteStream[0])
        nextStartCodeOffset = findNextStartCode(byteStream)
        nalUnit = byteStream[:nextStartCodeOffset]
        if nalUnitType == SPS:
            pv.consumeSps(nalUnit)
        elif nalUnitType == PPS:
            pv.consumePps(nalUnit)
        elif nalUnitType in (IDR_SLICE, SLICE):
            totalBytes += len(nalUnit) - 1
            frameBytesConsumed = pv.consumeSliceNalUnit(nalUnit)
            perFrameBytesConsumed.append(frameBytesConsumed)
            bytesConsumed += frameBytesConsumed
        byteStream = byteStream[nextStartCodeOffset:]
pvScores = pv.getScores()
assert bytesConsumed <= (totalBytes * 0.02), “Byte count exceeded”
return (pvScores, perFrameBytesConsumed)
```

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
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	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
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, and associated measurements and tests
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, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2017