International Telecommunication Union

ITU-T

P.1204.4

(01/2020)

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

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

Models and tools for quality assessment of streamed media

Video quality assessment of streaming services over reliable transport for resolutions up to 4K with access to full and reduced reference pixel information

Recommendation ITU-T P.1204.4

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	P.10–P.19
Voice terminal characteristics	P.30–P.39
Reference systems	P.40–P.49
Objective measuring apparatus	P.50–P.59
Objective electro-acoustical measurements	P.60–P.69
Measurements related to speech loudness	P.70–P.79
Methods for objective and subjective assessment of speech quality	P.80–P.89
Voice terminal characteristics	P.300–P.399
Objective measuring apparatus	P.500–P.599
Measurements related to speech loudness	P.700–P.709
Methods for objective and subjective assessment of speech and video quality	P.800–P.899
Audiovisual quality in multimedia services	P.900–P.999
Transmission performance and QoS aspects of IP end-points	P.1000–P.1099
Communications involving vehicles	P.1100–P.1199
Models and tools for quality assessment of streamed media	**P.1200–P.1299**
Telemeeting assessment	P.1300–P.1399
Statistical analysis, evaluation and reporting guidelines of quality measurements	P.1400–P.1499
Methods for objective and subjective assessment of quality of services other than speech and video	P.1500–P.1599

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

Recommendation ITU-T P.1204.4

Video quality assessment of streaming services over reliable transport for resolutions up to 4K with access to full and reduced reference pixel information

Summary

Recommendation ITU-T P.1204.4 describes the reduced-reference and full-reference video quality estimation model for Recommendation ITU-T P.1204 for monitoring the video quality for streaming using reliable transport (e.g., hypertext transfer protocol- (HTTP-) based adaptive streaming (HAS) over the transmission control protocol (TCP), quick user datagram protocol internet connections (QUIC)). The estimate is validated for videos encoded with H.264, H.265 or video payload type 9 (VP9) codecs at any resolution up to 4K/ultra-high definition (UHD) (3 840 × 2 160) resolution for personal computer (PC) monitors and television (TV) and up to 2 560 × 1 440 for smartphone and tablet displays.

The ITU-T P.1204 series of Recommendations provides sequence-related (between 5 s and 10 s) and per-1-second video-quality estimation. In principle, the per-one-second outputs of these video-quality models can be used together with an audio model for integration into audiovisual quality and, together with information about initial loading delay and media playout stalling events, further into a final per-session model output, an estimate of integral per-session quality (see e.g., Recommendations ITU-T P.1203, ITU-T P.1203.2, ITU-T P.1203.3).

Recommendation ITU-T P.1204.4 was developed in collaboration with the Video Quality Experts Group (VQEG).

This ITU-T P.1204-series of Recommendations addresses three application areas:

- large-screen presentation as with fixed-network video streaming;
- mobile streaming on handheld devices such as smartphones;
- presentation on tablet-type devices.

History

Edition	Recommendation	Approval	Study Group	Unique ID[*]
1.0	ITU-T P.1204.4	2020-01-13	12	11.1002/1000/14157

Keywords

Adaptive streaming, IPTV, mean opinion score (MOS), mobile video, mobile TV, monitoring, multimedia, OTT, 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

Page

Recommendation ITU-T P.1204.4

Video quality assessment of streaming services over reliable transport for resolutions up to 4K with access to full and reduced reference pixel information

1 Scope

This Recommendation describes a full-reference/reduced-reference video quality assessment module that together with audio and integration modules can be used to form a complete model to predict the impact of audio and video media encodings and observed Internet protocol (IP) network impairments on quality experienced by the end-user in multimedia streaming applications. The streaming techniques addressed comprise progressive download and adaptive streaming, for both mobile and fixed network streaming applications. The video quality module can also be used stand-alone as a video quality prediction model.

The model described here is a pixel-based reduced-reference model, which can be seen as a special form of a full-reference model. It can be used from monitoring to accurate measurement of client-side quality of experience (QoE) for network or service optimization or benchmarking purposes. The model developed may be deployed both in end-point locations and at mid-network monitoring points, as long as the reference information can be made available at that location. The model in this Recommendation can also be used for laboratory testing of video systems.

The model described here is applicable to progressive download and adaptive streaming or other streaming applications with reliable transport, where the quality experienced by the end user is affected by video degradations due to coding, spatial re-scaling or variations in video frame rates. Quality assessment of adaptive streaming includes aspects of media adaptation that may be handled in integration modules such as [ITU-T P.1203.3], and not in the video modules in this Recommendation. This Recommendation is able to handle various video codecs (i.e., H.264, H.265/high-efficiency video coding (HEVC), and video payload type 9 (VP9), resolutions up to 4K/ ultra-high definition-1 (UHD-1) and frame rates up to 60 frames/s. The video-quality module Pv of [b-ITU-T P.1203], i.e., [ITU-T P.1203.1], only addresses H.264 and full high definition (HD) with up to 30 frames/s.

The model predicts a mean opinion score (MOS) on a five-point absolute category rating (ACR) scale (see [ITU-T P.910]) as an overall video quality MOS (5 s to 10 s). In addition to the overall quality score, this video quality model produces a per-one-second quality score, suitable for diagnostics or integration into an integral quality score for longer sessions (see, for example, [ITU-T P.1203.3] for 1 min to 5 min duration sessions).

The model associated with this Recommendation cannot provide a comprehensive evaluation of the video quality as perceived by an *individual end-user* because the scores reflect the perceived impairments due to coded video media data being transmitted over an IP connection with certain performance and do not include specific terminal device or user-specific information. The scores predicted by such a general quality model necessarily reflect *average perceptual quality*.

Effects due to source generations, such as signal noise, video shake, certain colour properties (and other similar video factors) and other impairments related to the payload, are not reflected in the scores computed by this model.

As a consequence, this Recommendation can be used for applications such as:

– in-service quality monitoring for specific IP-based audiovisual services, as specified in more detail in clause 6.1;

– performance and quality assessment of live networks (including video encoding) considering the effect due to encoding bitrate, encoding resolution and encoding frame rate;

- laboratory testing of video systems;
- benchmarking of different service implementations;
- benchmarking of different encoder implementations;
- evaluation of transcoding solutions.

In particular, targeted applications are progressive download streaming and adaptive streaming (using reliable transport), which includes the following.

- Over-the-top (OTT) services, as well as operator-managed video services (over the transmission control protocol (TCP)).
- Video over both mobile and fixed connections.
- The streaming protocols HTTP live streaming (HLS) or dynamic adaptive streaming over HTTP (DASH) used with the hypertext transfer protocol (HTTP) or HTTP2 over TCP/IP or quick user datagram protocol internet connections (QUIC), or real-time messaging protocol (RTMP) over TCP/IP. Note that the model is agnostic to the specific application or transport layer protocol, with the exception that it assumes reliable delivery of video packets.
- Video services typically using container formats based on the ISO/IEC base media file format such as Moving Picture Experts Group-4 (MPEG-4) Part 14 (MP4), or other container formats such as audio video interleave (AVI), Matroska video (MKV), WebM, Third Generation Partnership (3GP), and MPEG-2 transport stream (MPEG2-TS). Note that the model is agnostic to the type of container format.

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 (2019), *Advanced video coding for generic audiovisual services*.

[ITU-T H.265] Recommendation ITU-T H.265 (2019), *High efficiency video coding*.

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

[ITU-T P.1203.1] Recommendation ITU-T P.1203.1 (2019), *Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport – Video quality estimation module.*

[ITU-T P.1203.3] Recommendation ITU-T P.1203.3 (2019), *Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport – Quality integration module.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following term defined elsewhere:

3.1.1 bitstream [ITU-T H.264]: A sequence of bits that forms the representation of coded pictures and associated data forming one or more coded video sequences. Bitstream is a collective term used to refer either to a NAL unit stream or a byte stream.

3.1.2 integral quality [b-ITU-T P.1203]: The quality as perceived by a subject in a subjective test, which corresponds to the scope of this Recommendation. Artefacts presented in the subjective tests typically include a combination of audio compression, video compression, and stalling effects.

3.1.3 mean opinion score (MOS) [b-ITU-T P.1204]: The mean of opinion scores, which are values on a predefined scale that subjects assign to their opinion of the performance of the telephone transmission system used either for conversation or for listening to spoken material.

NOTE – Paraphrased from clause 7 of [b-ITU-T P.800.1].

3.1.4 media adaptation [b-ITU-T P.1203]: Events where the player switches video playback between a known set of media quality levels while adapting to network conditions, by downloading and decoding individual segments in sequence.

3.1.5 media quality level [b-ITU-T P.1203]: A particular encoding setting applied to a video or audio stream.

3.1.6 model, model algorithm [b-ITU-T P.1203]: An algorithm with the purpose of estimating the subjective (perceived) quality of a media sequence.

3.1.7 sequence [b-ITU-T P.1203]: An audiovisual stream composed of multiple non-overlapping segments.

3.1.8 video chunk [b-ITU-T G.1022]: A contiguous set of samples for one track of a video.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

3GP	Third Generation Partnership
ACR	Absolute Category Rating
AV1	AOMedia Video 1
AVC	Advanced Video Coding
AVI	Audio Video Interleave
DASH	Dynamic Adaptive Streaming over HTTP
GoP	Group of Pictures
HAS	HTTP-based adaptive streaming
HD	High Definition
HEVC	High-Efficiency Video Coding
HLS	HTTP live streaming
HTTP	Hypertext Transfer Protocol

IP	Internet Protocol
MKV	Matroska Video
MO	Mobile
MOS	Mean Opinion Score
MP4	MPEG-4 Part 14
MPEG	Moving Pictures Expert Group
MPEG2-TS	MPEG-2 Transport Stream
PC	Personal Computer
QoE	Quality of Experience
QUIC	Quick User datagram protocol Internet Connections
RMSE	Root Mean Square Error
RTMP	Real-Time Messaging Protocol
RTP	Real-time Transport Protocol
TA	Tablet
TCP	Transmission Control Protocol
TS	Transport Stream
UDP	User Datagram Protocol
UHD	Ultra-High Definition
VVC	Versatile Video Coding

5 Conventions

This Recommendation uses the following conventions:

- 4K: Video resolution of 4 096 × 2 160 or 3 840 × 2 160;
- Pv designates the video quality estimation module (as specified in this Recommendation for the case of full and reduced reference pixel-based prediction, see [b-ITU-T P.1204] for alternative implementations such as bitstream based and hybrid);
- Reliable transport: Reliable delivery with protocols guaranteeing no loss of information.

6 Areas of application

6.1 Application range for the models

Table 1 shows the application range of the model in this Recommendation based on what the model has actually been developed for. Table 2 lists areas where it is not applicable. Table 3 lists test factors and coding technologies for which this Recommendation has been validated.

Table 1 – Application areas for which this Recommendation is applicable

Areas for which the model is applicable
In-service monitoring of video sent over reliable transport. Both OTT services and operator-managed video services, using reliable delivery with protocols such as HTTP or HTTP2 over TCP/IP or QUIC, or RTMP over TCP/IP. Note that this model is agnostic to the type of container format.
Performance and quality assessment of live networks (including video encoding) considering impairments due to encoding bitrate, encoding resolution, and encoding frame rate.
Laboratory testing of video systems.
Benchmarking of different service implementations.
Benchmarking of different encoder implementations. Note that only the full and reduced reference pixel-based model type can be used for direct benchmarking of this type.
Evaluation of transcoding solutions.

Table 2 – Application areas for which this Recommendation is not applicable

Areas for which the model is not applicable
In-service monitoring of video streaming using unreliable transport (e.g., real-time transport protocol/user datagram protocol (RTP/UDP), where packet loss introduces visible quality degradations.
Evaluation of visual quality of display/device properties.
Evaluation of audio/video sync distortions.
Evaluation of video codecs for which the model is not validated (AOMedia Video 1(AV1), MPEG-I Part 3 [versatile video coding (VVC)], etc.).
Evaluation of the effects of noise, delay, colour correctness or other content-production-related aspects.

Table 3 –Test factors, and coding technologies for which this Recommendation has been validated

Video test factors for which the model has been validated	
Video content	Movies and movie trailers, sports videos, documentaries, computer-generated graphics/games, etc.
Input video length	The video modules were trained and validated to produce one overall video-quality score for a chunk of ~7-9 s and also provide the per-second scores. Optimal performance for ~8 s. Models are assumed to provide valid overall video-quality estimations for 5-10 s long sequences.
Bitstream container	AVI, MP4, MKV, WebM
Encoder types (and implementation), see Note 1	H.264/advanced video coding (AVC) (libx264), H.265/HEVC (libx265), VP9 (libvpx-vp9)

Table 3 –Test factors, and coding technologies for which this Recommendation has been validated

Video test factors for which the model has been validated				
Encoder profiles	H.264 (MPEG-4 Part 10): Constrained baseline, Main, Hi, Hi10, Hi422. H.265: Main, Main10, Rext. VP9: 0, 1, 2, 3.			
Video resolution and bitrate	**Resolution definition**	**Video height range**	**Personal computer/television (PC/TV)**	**Mobile/tablet (MO/TA)**
	Below SD	180-70	–	90 Kbps – 1 Mbps
	SD	360-540	150 Kbps – 4 Mbps	150 Kbps – 4 Mbps
	HD	720-1 080	500 Kbps – 15 Mbps	500 Kbps – 15 Mbps
	Above HD	1 440-2 160	1.5 Mbps – 45 Mbps	1.5 Mbps – 20 Mbps
Video aspect ratio	16:9, see Note 2			
Group of pictures (GoP)	Variable. Average GoP length can be between 0.5 s and chunk duration			
Bit-depth	8 bits or 10 bits			
Chroma subsampling	YUV 4:2:0 and YUV 4:2:2			
OTTs	Online providers that offer video on demand and video encoding as a service. It should be noted that the models are applicable for similar OTTs.			
Display resolution and frame rate	PC/TV; 2 160p, up to 60 frames/s. Mobile/Tablet: 1 440p, up to 60 frames/s.			
Viewing distances	PC/TV: 1.5*H* to 3*H* (*H*: Screen height), see Note 3 MO/TA: 4*H* to 6*H*			
NOTE 1 – During training and validation, FFmpeg 3.2.2 was used with x264 snapshot 20170202-2245, x265 v2.2, libvpx 1.6.1. NOTE 2 – For original content with a larger aspect ratio, letterboxing of up to 30% was allowed, that is 1512 pixels height for video coded at 2160 pixels height. Video content with 1.89:1 aspect ratio (e.g., cinema 4K) may also be used. NOTE 3 – It is noted that for PC/MO, the model output is conservative and should be interpreted to correspond to a viewing distance of 1.5*H* to 1.6*H*.				

7 Video module in the ITU-T P.1204 context

The building blocks of the ITU-T P.1204 model used standalone are depicted in Figure 1.

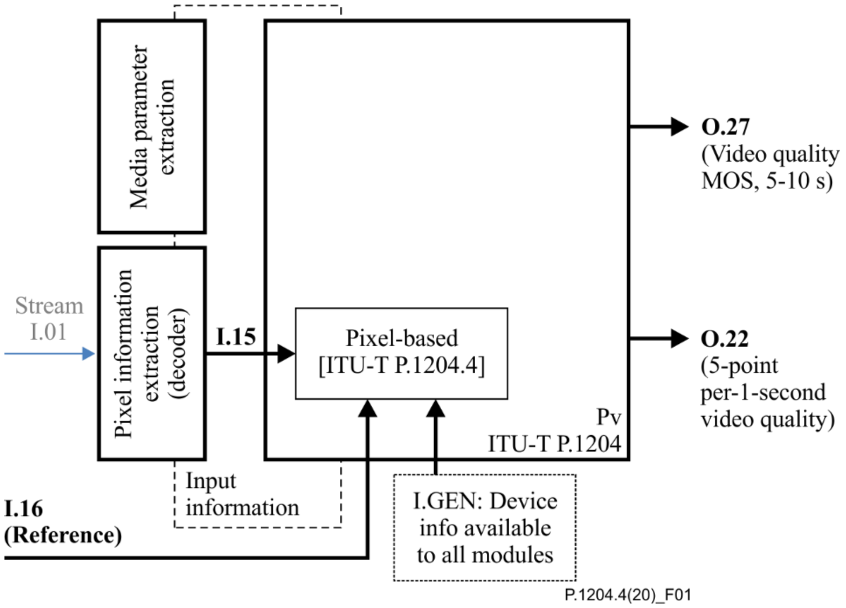
Media parameter extraction
Stream
I.01
Pixel information extraction (decoder)
I.15
Pixel-based
[ITU-T P.1204.4]
Pv
ITU-T P.1204
O.27
(Video quality MOS, 5-10 s)
O.22
(5-point per-1-second video quality)
I.16
(Reference)
Input information
I.GEN: Device info available to all modules
P.1204.4(20)_F01

Figure 1 – Building blocks of the ITU-T P.1204 video quality model used stand-alone in the case of this Recommendation

8 Model input

In a full-reference model, quality q of a test video v – called a degraded video – is estimated by a function G depending on the degraded video v and on the reference video v_{ref},

$$q = G(v, v_{\mathrm{ref}})$$

In the reduced-reference case, the function G depends on the reference through features f_{ref} of the reference v_{ref} only. The features are extracted by the reference feature extraction function ϕ,

$$f_{\mathrm{ref}} = \phi(v_{\mathrm{ref}})$$

and there is a restriction on the size of the features. The quality of the degraded video is estimated by function G' by

$$q = G'(v, f_{\mathrm{ref}}).$$

Thus, a reduced-reference model is a special form of a full-reference model. The reference features f_{ref} are sometimes called the *side information*, as in an operational setup this information can be transmitted over a side-channel to the measurement device.

The model receives information **I.GEN** about the viewing condition per media session: *display resolution, device type, display size,* and *relative viewing distance*.

The video quality model receives the following input signals: the reference video for extraction of the reference features, and the degraded video signal, decoded and possibly upscaled. It is possible to process the video in chunks, respecting the sampling of the output signal. See Table 4 for an overview of model inputs.

Table 4 – Input description of I.GEN, I.15, I.16

ID	Description	Values	Frequency
I.GEN			
0	The resolution of the image displayed to the user	Number of pixels (*W* × *H*) in displayed video	Per media chunk

Table 4 – Input description of I.GEN, I.15, I.16

ID	Description	Values	Frequency
1	The device type on which the media is played	"PC", "TV", "MO", "TA"	Per media chunk
2	Device display size	Diagonal size (diagonal in inches)	Per media chunk
3	Relative viewing distance in multiple of display height	Relative viewing distance	Per media chunk
I.15			
4	Degraded video	The raw pixels (YUV file including metadata required for parsing; width, height, frame rate, and pixel format) of the processed video, i.e., the video decoded and upscaled to display resolution without buffering or stalling. The frame information in I.16 and I.15 is synchronized, i.e., no frame misalignments are present.	Per media chunk
I.16			
5	Reference video information	The reference-side information extraction module takes as input the reference video and outputs the side information file. The reference model side channel bandwidth limit is 256 kbit/s. Thus, the side information of the reference model for a video sequence *v* is stored in a file with size at most 256/8 * t_{v} kB, where t_{v} is the duration of video v in seconds.	Per media chunk

9 Model algorithm and output

The video module defined in this Recommendation has two outputs, O.22 and O.27. It provides output values on the five-point ACR scale (MOS).

10 Model description

This clause describes the video model in detail.

10.1 Notation

Variable names are single or multiple characters of the form $frame\_height$ or using subscripts like $n_{\mathrm{resolution}}$, to shorten formulas. For an array, x, the notations (x) and $(x_{ij})_{i=0..m-1,j=0..n-1}$ are used. The elements are denoted x_{ij} or $x[i,j]$, where array locators start at 0. The shorthand $x[i]$ is used for $x[i,\cdot]$.

A video sequence is given by a triple (Y, Cb, Cr) of sequences of planes of YC_bC_r pixel values, like in e.g., YUV420p, together with a sequence of display times (dt) for each frame. The variables $frame\_height, frame\_width$ refer to the $height, width$ of the Y plane, and n_{frame} denotes the number of frames of the video sequence.

Model constants and parameters are given in the lists at the end of the descriptions. Most clauses describe a set of computations, with input and output variables given at the start. In addition, the dimensions of the variables are given in the rightmost column. Some of these dimensions are given by model constants, e.g., $n_{\text{resolution}}$, their values can be found in the list at the end.

10.2 Multi-resolution frame pyramid

INPUT:

Name	Description/dimension
Y, Cb, Cr	Sequence of *Y*, *Cb*, *Cr* planes, pixel values of the video signal
dt	number of frames, n_{frame}

OUTPUT:

Name	Dimension
$(Y_{l,r})$	$n_{\text{frame}} \times n_{\text{resolution}} \times \sum_i (frame\_height_i \times frame\_width_i)$

All video frames are read and subsampled to a multi-resolution pyramid as follows: The *Y* planes of frames are read and re-scaled by bicubic interpolation to a fixed resolution of $(f\_height\_init, f\_width\_init)$ and scaled to the range $[0, y\_range\_max]$. As a reference ffmpeg version 3.2.2 can serve, see [b-FFmpeg]. To each frame a display time dt is associated, e.g., for a video at a constant frame rate of 25 frames/s, the display time (dt) is the constant vector of 40 ms for each frame.

Frames are iteratively down-scaled by smoothing the frame data by convolution with filter kernel $f = [\frac{1}{4}, \frac{1}{2}, \frac{1}{4}]$, keeping the same size of the matrix, computing the convolution by extending the border with constant values, and subsampling by 2 in either spatial direction, resulting for each frame F_l in a frame multi-resolution pyramid of the *Y* data given by

$$(Y_{l,0}, Y_{l,1}, .., Y_{l,N-1}), \tag{1}$$

with $N = n_{\text{resolution}}$. The dimension of $Y_{l,r}$ is

$$\dim(Y_{l,r}) = \frac{f\_height\_init}{2^{N-1-i}} \times \frac{f\_width\_init}{2^{N-1-i}} \tag{2}$$

10.3 Frame edge representation

INPUT:

Name	Dimension
$(Y_{l,r})$	$n_{\text{frame}} \times n_{\text{resolution}} \times \sum_i (frame\_height_i \times frame\_width_i)$
dt	n_{frame}

OUTPUT:

Name	Dimension
Z	$n_{\text{frame}} \times n_{\text{resolution}} \times \sum_i (frame\_height_i \times frame\_width_i)$
ϕ	$n_{\text{frame}} \times n_{\text{resolution}} \times \sum_i (frame\_height_i \times frame\_width_i)$

Features based on edge orientation and strength will be computed for all $Y_{l,r}$. To simplify notation in the following paragraph, the subscripts are omitted, and Y signifies a $Y_{l,r}$ for some values of l, r. Start by computing edge strength R and edge orientation ϕ in the following way: the horizontal edges H are computed as

$$H[i,j] = \frac{2}{\pi} \cdot \arctan\left(\frac{1}{y_{\mathrm{rescale}}} (Y[i,j] - Y[i-1,j])\right) \tag{3}$$

and accordingly

$$V[i,j] = \frac{2}{\pi} \cdot \arctan\left(\frac{1}{y_{\mathrm{rescale}}} (Y[i,j] - Y[i,j-1])\right) \tag{4}$$

where the locators i, j run from 0 to $\dim(Y) - 1$. A polar representation is computed composed of edge strength R,

$$R[i,j] = \sqrt{H[i,j]^2 + V[i,j]^2} \tag{5}$$

and angle ϕ as

$$\phi[i,j] = \mathrm{angle}(H[i,j], V[i,j]) \tag{6}$$

where the angle is measured in the counterclockwise direction, in the range $[0,2\pi]$. Further, a normalized edge strength with lateral inhibition is computed.

Compute c as the average of the constant c_{lat} and the average of R,

$$c = \frac{1}{2} \cdot (c_{lat} + E(R)), \tag{7}$$

here E is the average computed over all spatial position locators i, j.

For a position locator (i,j), a pair of positions (i_0, j_0) and (i_1, j_1) perpendicular to the edge are defined by

$$i_0 = \frac{\Delta_{lat} \cdot H[i,j]}{\max(d_{lat}, R[i,j])} \tag{8}$$

$$j_0 = \frac{\Delta_{lat} \cdot V[i,j]}{\max(d_{lat}, R[i,j])} \tag{9}$$

and

$$i_1 = -i_0, \quad j_1 = -j_0. \tag{10}$$

A normalization array S by lateral inhibition is given by the edge strength at positions perpendicular to the edge

$$S[i,j] = 0.5 * (Z[i + r(i_0), j + r(j_0)] + Z[i + r(i_1), j + r(j_1)]) \tag{11}$$

where the function r rounds to the next integer value. The normalized edge strength Z is computed as

$$Z[i,j] = \frac{\max(0, R[i,j] - S[i,j])}{c + R[i,j] + S[i,j]} \tag{12}$$

10.4 Local edge statistic

INPUT:

Name	Dimension
Z	$n_{\mathrm{frame}} \times n_{\mathrm{resolution}} \times \sum_i (frame\_height_i \times frame\_width_i)$

ϕ	$n_{\text{frame}} \times n_{\text{resolution}} \times \sum_{i} (frame\_height_i \times frame\_width_i)$

OUTPUT:

Name	Dimension
s	$n_{\text{frame}} \times n_{\text{resolution}} \times n_{\text{orient}} \times ns_h \times ns_w$

A partition of unity is a family of positive continuous [0,1]-valued functions $(\theta_k)_{k=0,,,L-1}$ for some integer L having

$$\sum_{k=0}^{L-1} \theta_k = 1. \tag{13}$$

A partition of unity on the unit circle is given by the functions $(\theta_k)_{k=0,..L_o-1}$ mapping an angle ϕ in [0,2π] to

$$\theta_k(\phi) = 1 \quad \text{if} \quad \Delta(\phi, \alpha_k) < \beta \tag{14}$$

further

$$\theta_k(\phi) = \frac{2\beta - \Delta(\phi, \alpha_k)}{\beta} \quad \text{if} \quad \beta < \Delta(\phi, \alpha_k) < 2\beta \tag{15}$$

and

$$\theta_k(\phi) = 0 \quad \text{if} \quad \Delta(\phi, \alpha_k) >= 2\beta \tag{16}$$

with $\Delta(\phi, \alpha_k) = \min(|\phi - \alpha_k|, 2\pi - |\phi - \alpha_k|)$. Here $\alpha_k = k \cdot 2\pi/L_{\text{o}}$ and $\beta = 2\pi/(3L_{\text{o}})$.

Let v_δ denote the continuous partial linear function given by

$$v_\delta(x) = \frac{1}{2} \cdot \frac{x}{\delta} \quad 0 <= x < \delta \tag{17}$$

$$v_\delta(x) = \frac{1}{2} \quad \delta <= x < 2\delta \tag{18}$$

$$v_\delta(x) = \frac{1}{2}(3 - \frac{x}{\delta}) \quad 2\delta <= x < 3\delta \tag{19}$$

$$v_\delta(x) = 0 \quad otherwise \tag{20}$$

A partition of unity Ψ_{mn} in the spatial domain of the frame data Y is defined by

$$\Psi_{mn}[i,j] = v_\delta(i - m) * v_\epsilon(j - n) \tag{21}$$

with $\delta = height(Y)/(n_h + 2)$ and $\epsilon = width(Y)/(n_w + 2)$.

A family of patches (P_{mnk}) is computed using the partition of unity Ψ_{mn} in the spatial domain and the partition (θ_k) in orientation. For orientation subscript k, and locators i,j a local patch P is computed by

$$P_{mnk}[i,j] = \Psi_{mn}[i,j] \cdot Z_{mn}[i,j] \cdot \theta_k(\phi_{mn}[i,j]) \tag{22}$$

The local patch P_{mnk} is used to compute a patch statistic: Let $(z_{mnk}[i])_{i=0,..,N-1}$ denote the entries of the local patch P_{mnk} sorted in ascending order, then a statistic s is defined as the average over all values of z_{mnk} above the quantile q,

$$s_{mnk} = s(z_{mnk}) = \frac{1}{N - q} \sum_{i=q}^{N-1} z_{mnk}\,[i] \tag{23}$$

with $q = floor((width(Y) - q_{pos})/width(Y))$. The patch statistic s_{mnk} computed based on the lth frame $Y_{l,r}$ at resolution r is denoted by $s_{mnk}^{l,r}$ if the reference to the frame number and resolution index is needed, otherwise, if clear from the context, the simpler notation s_{mnk} is used, as in Formula (23).

For each subsampled frame and each resolution, the local patch statistic $s = (s_{mnk})$ is computed. To reduce the number of local statistics, a border of 2 is skipped, and the subscripts m, n are sampled by 2, e.g., sampled at positions 2,4,6,…, thus resulting in dimension of $dim(s) = number_{\text{frame}} \times number_{\text{resolution}} \times n_{\text{orient}} \times ns_h \times ns_w$.

To speed up computation, the local statistics are computed for a subset only. The patch statistic $s_{mnk}^{l,r}$ for frame l at resolution r is computed only if $r < num_{h\_res}$ or the following two conditions are fulfilled, otherwise it is set to zero. The first is that the following frame sampling condition is met, $(l \mod fps) \mod m = 0$, where fps is the video frame rate and m is the largest integer less than or equal to $fps/num_{high\_res\_per\_chunk}$. The second condition is that the local patch statistic at lower resolution is above the average overall patch statistics at lower resolution,

$$s_{mnk}^{l,r-1} > E(s^{l,r-1}), \tag{24}$$

where the average E is computed over all local patches in space, $m = 0, .., ns_h - 1$, $n = 0, .., ns_w - 1$, and orientation, $k = 0, .., n_{orient} - 1$.

10.5 Sharpness

INPUT:

Name	Dimension
s	$ns_{\text{frame}} \times n_{\text{resolution}} \times n_{\text{orient}} \times ns_h \times ns_w$

OUTPUT:

Name	Dimension
$sharp$	ns_{frame}

At the highest resolution $r = number\_resolution - 1$ of the pyramid and for each frame l with local edge statistic $(s_{mnk}^{l,r})_{mnk}$, let $(h_i)_{i=0,..,N}$ denote the values of $(s_{mnk}^{l,r})_{mnk}$ in ascending order. Then sharpness $sharp$ is computed as the mean over the fraction of the largest $sharp\_frac$ values of h,

$$sharp = \frac{1}{N - i_0 + w_0} \cdot \left(w_0 \cdot h[i_0] + \sum_{i=i_0+1}^{N-1} h\,[i] \right) \tag{25}$$

Here $i_0 = floor(N \cdot sharp\_frac)$ and $w_0 = 1 - (N \cdot sharp\_frac - i_0)$.

The computed sharpness value is scaled by multiplication with *sharp_scale_fac* and stored as a 16-bit float.

10.6 Features of reference (side information)

INPUT:

Name	Dimension
Y, Cb, Cr	Sequence of *Y*, *Cb*, *Cr* planes, pixel values of the video signal
dt	n_{ref_frame}

OUTPUT:

Name	Dimension

s_{ref}	$ns_{\text{ref\_frame}} \times n_{\text{orient}} \times ns_h \times ns_w$
$sharp$	$ns_{\text{ref\_frame}}$

Starting from the reference video sequence, the multi-resolution frame pyramid $(Y_{l,r})$, the frame edge representation, the local edge statistic $s^{l,r}$ and sharpness $sharp$ are computed.

Set $fs_{\text{step}} = 1$, but for frame rates higher than 30 frames/s, set $fs_{\text{step}} = 2$. The values of $s^{l,r}$ will be sampled by fs_{step} , resulting in $ns_{\text{ref\_frame}} = floor(n_{\text{ref\_frame}}/fs_{\text{step}})$ many frames with corresponding local statistics, in addition, the statistics will be kept at one resolution, only. In more detail, for each frame, the local orientation edge statistics s at resolution $dissim\_res$ is sampled by fs_{step}, to reduce the number of statistics to keep, thus

$$s_{\text{ref}}[l, m, n, k] = s[l \cdot fs_{\text{step}}, dissim\_res, m, n, k] \tag{26}$$

with $i = 0, .., ns_h - 1, j = 0, .., ns_w - 1, k = 0, .., n_{orient} - 1$. Note that for clarity the subscript $_{\text{ref}}$ is used to denote the statistic of the reference video.

The extracted reference features are composed of the local edge statistics s_{ref}, compressed by multiplying the values by $stat\_scale\_fac$, rounded to integer, clipped to [0,255] and stored as a uint8, together with display_time dt and sharpness sharp, where the latter two are stored as float16 values.

10.7 Features of test video

INPUT:

Name	Dimension
Y, Cb, Cr	Sequence of Y, Cb, Cr planes, pixel values of the video signal
dt	n_{frame}

OUTPUT:

Name	Dimension
s	$ns_{\text{frame}} \times n_{\text{resolution}} \times n_{\text{orient}} \times ns_h \times ns_w$
rep_frame	ns_{frame}
y_low_res	$ns_{\text{frame}} \times y\_low\_res\_height \times y\_low\_res\_width$
$sharp$	ns_{frame}

Starting from the test video sequence, the multi-resolution frame pyramid $(Y_{l,r})$, the frame edge representation, the local edge statistic $s_{mnk}^{l,r}$ and sharpness $sharp$ are computed. To speed up computation, the values of $s_{mnk}^{l,r}$ are computed for $l = 0, d, 2d, \ldots$ for a frame sampling step d defined by the test video frame rate as given in the following list:

Step d	Video frames per second (fps)
4	fps > 30
2	20 < fps <= 30
1	<=20

Hence, $ns_{\text{frame}} = floor(n_{\text{frame}}/d)$.

To match the features of the reference, the same compression is applied to the values of $s_{mnk}^{l,r}$ to be stored as uint8 values, and sharpness $sharp$ as float16.

A vector rep_frame of indicators for repeated frames is computed as follows. If frame i is a repetition of frame $i-1$, then $rep\_frame[i] = 1$, otherwise $rep\_frame[i] = 0$.

For a sequence of matrices $(x_l)_{n=0,\ldots,L-1}$, each of dimension $M \times N$, the spatial local mean (z_l) is a sequence of dimension $L \times M_0 \times N_0$ given by

$$z_l[p,q] = E(x_l[i,j]) \qquad (27)$$

where the average E is computed over all $i = floor(p * M/M_0), \ldots, floor((p+1) * M/M_0$, and $j = floor(q * N/N_0, \ldots, floor((q+1) * N/N_0)$. Accordingly, starting from the Y values at the lowest resolution in the pyramid, $Y_{l,0}$, an array of local average values *y_low_res* is computed using

$$y\_low\_res_l[p,q] = E(Y_{l,0}), \qquad (28)$$

with $p = 0, .., y\_low\_res\_height - 1$, and $q = 0, .., y\_low\_res\_width - 1$.

10.8 Additional features

INPUT:

Name	Dimension
s	$ns_{\text{frame}} \times ns_{\text{resolution}} \times n_{\text{orient}} \times ns_h \times ns_w$
s_{ref}	$ns_{\text{ref\_frame}} \times n_{\text{orient}} \times ns_h \times ns_w$
dt	ns_{frame}
dt_{ref}	ns_{frame}
rep_frame	ns_{frame}
$sharp$	ns_{frame}
$sharp_{ref}$	$ns_{\text{ref\_frame}}$

OUTPUT:

Name	Dimension
$dissim$	$ns_{\text{frame}} \times ns_h \times ns_w$
$dissim_{inc}$	$ns_{\text{frame}} \times ns_h \times ns_w$
$motion_{avg}$	ns_{frame}
fps	ns_{frame}
$sharp$	ns_{frame}
$sharp_{ref}$	ns_{frame}

First, a vector of indices $I_{\text{ref}} = (I_{\text{ref},i})_i$ is determined, such that frame i of the test video corresponds to frame $I_{ref,i}$ of the reference video. The procedure is done to possibly support frame rate reductions of the test video with respect to the reference. First, a matrix A of root mean square error (RMSE) values is computed,

$$A_{ij} = \text{RMSE}(s[i, dissim\_res], s_{ref}[j]). \qquad (29)$$

Then, the index is determined as the best match close to a regression line estimated by robust regression. The details are given by the following pseudo code, the resulting index is I_{ref} = *ref_ind_reg(A)*.

```
function arg_min_constraint(v,x,delta=2):
  # Determine the position of the minimum within [x-delta,x+delta].
  # INPUT: v -- numpy 1-d array, x -- float, delta -- int.
  # OUTPUT: i -- int, position of minimum under constraint 'close to x'.
```

```
    i0 = clip( int(ceil(x-delta)), 0, len(v)-1)
    i1 = clip( int(floor(x+delta)), 1, len(v))
    return i0 + argmin(v[i0:i1])


  function ref_ind_reg(A):
    # INPUT: A -- numpy array (2D) with dim m x n, of dissimilarity values.
    # OUTPUT: ref_ind -- numpy array (1-dim) of length m.

    # determine a starting point
    for i in [0,1,..,m-1]:
      x[i] = arg_min_constraint(A[i,:], floor(i*n/m),delta=6)
    r = ref_ind(A)

    # limit outliers using a Huber regression
    TA = sklearn.linear_model.HuberRegressor()
    TA.fit(x,r)
    r_estim = TA.predict(x)
    # take as ref ind values close by 'r_estim'
    for i in [0,..m-1]:
      r_c[i] = arg_min_constraint(A[i,:],r_estim[i])

    return r_c
```

For more details of the Huber robust regression, see [b-Scikit-learn], [b-Huber].

Let the function D compute for two arrays $(a[k,i,j])$, $(b[k,i,j])$ the average positive part of the difference,

$$D(a,b) = \sum_{k} \max(0,a-b) / \sum_{k} \mathrm{sign}(\max(0,a-b)) \tag{30}$$

where the max and sign functions are applied element-wise. Then, two arrays of dissimilarities $dissim$, $dissim_{inc}$ are computed by

$$dissim[i] = D(s[i,dissim\_res], s_{ref}[i]) \tag{31}$$

and

$$dissim_{inc}[i] = D(s_{ref}[i], s[i,dissim\_res]) \tag{32}$$

An average motion feature is computed by

$$motion_{\mathrm{avg}}[i] = E(\mathrm{abs}(s[i,dissim_{res}] - s[\max(0,i-1), dissim_{res}]) \tag{33}$$

where abs denotes the absolute values and E denotes the mean computed over all $n_{\mathrm{orient}} \times ns_h \times ns_w$ array entries of s.

A windowed frame rate is computed as follows. The vector of frame display times is split into chunks of fps_chunk_dur seconds duration with chunk start and end given by $(start_i, end_i)$, where the last chunk is allowed to be longer, if the video duration is not a multiple of fps_chunk_dur. A frame rate is computed for each chunk given by $(start_i, end_i)$ by computing

$$fps[i] = \frac{1000}{E(dt_{\mathrm{no\_rep}})} \qquad i \quad \text{in} \quad [start_i, \dots, end_i] \tag{34}$$

where the function E computes the average, $dt_{\mathrm{no\_rep}}$ is the display time in milliseconds of each frame in the chunk ignoring repeated frames, i.e.,

$$dt_{\mathrm{no\_rep}}[j] = dt[j_0] + \dots + dt[j_m] \tag{35}$$

for j_0 being the jth non-repeated frame, and there are m consecutive repeated frames following frame j_0. A non-repeated frame is a frame l having $rep\_frame[l] = 0$. The frame rate vector is resampled to match the size of $dissim$ using the step function resampling approach described in clause 10.9.

The sharpness vector $sharp$ and $sharp_{ref}$ contain zeros reflecting the skipped computation of the corresponding local edge statistic. These missing values are estimated by applying the function *estimate_missing_by_local_average* given by the following pseudo-code.

```
function esimate_missing_by_local_average(x):
  # Input a 1-dim array of positive values, with some zero values
  #   corresponding to skipped computations
  # Return a 1-dim array with zero entries replaced by local averages.

  m = num_a
  i_non_z = non_zero(x) # return indices on non-zero elements of x
  n = int(ceil(len(x)/len(i_non_z)))
  z = zeros(len(x))
  for i in range(len(z)):
    j0 = max(0,i - m*n)
    j1 = min(j0+2*m*n,len(x))
    j0 = max(0,j1-2*m*n)
    x_local = x[j0:j1]
    j_non_z = non_missing(x_local)[0]
    z[i] = np.mean(x_local[j_non_z])
  if len(z)<num_s:
    z[i] = ones(len(x))*mean(z)
  return z
```

In addition, the sharpness vector $sharp_{\mathrm{ref}}$ of the reference video is resampled to the size ns_{frame} of vector $sharp$ using the step function resampling approach described in clause 10.9.

10.9 Step functions

For a vector $(v_i)_{i=0,..,N-1}$ and a monotonically increasing vector $(t_i)_{i=0,..,N}$ let $S_{v,t}\colon \mathbb{R} \rightarrow \mathbb{R}$ denote a step function, a real-valued function taking the constant value v_i over the intervall $[t_i, t_{i+1}]$ and zero otherwise. Let $\mathrm{Avg}(S_{v,t}, t_0, t_1)$ denote the average value of the step function $S_{v,t}$ over the interval $[t_0, t_1]$.

Given an additional monotonically increasing vector $(t'_i)_{i=0,...N}$, the step function $S_{v,t}$ can be resampled to the interval limits of (t'_i) to the step function $S_{v',t'}$ in a natural way by integrating the function $S_{v,t}$ over the intervals defined by (t'_i), explicitly

$$v'_i = \frac{1}{t'_{i+1} - t'_i} \int_{t'_i}^{t'_{i+1}} S_{v,t}(x)dx = \mathrm{Avg}(S_{v,t}, t'_i, t'_{i+1}) \tag{36}$$

10.10 Transformations

Let $S\colon \mathbb{R} \rightarrow [0,1]$ be a non-linear transformation composed of two parts, and parameterized by p_x the joining position of the two parts and its mapped-value $S(p_x) = p_y$ and slope p_q at p_x. The first part is given by

$$S(x) = a \cdot x^b \quad if \quad x <= p_x \tag{37}$$

and the second part by an exponential saturation of the form

$$S(x) = 2 \cdot d \cdot \left(\frac{1}{1 + \exp\big(-c \cdot (x[I] - px)\big)} - \frac{1}{2} \right) + p_y, \quad x > p_x \tag{38}$$

with

$$a = p_y/p_x^b, \quad b = p_x * p_q/p_y \quad c = 2 * p_q/d \quad d = 1 - p_y. \tag{39}$$

For most of the parameters that are used in the following, the transformation has an S-shape, and will therefore be called an S-shaped transformation. The list in clause 10.11 summarizes the transformations that will be used together with their parameters.

10.11 Core model

INPUT:

Name	Dimension
s	$ns_{\text{frame}} \times ns_{\text{resolution}} \times n_{\text{orient}} \times ns_h \times ns_w$
$dissim$	$ns_{\text{frame}} \times ns_h \times ns_w$
$dissim_{\text{inc}}$	$ns_{\text{frame}} \times ns_h \times ns_w$
$motion_{\text{avg}}$	ns_{frame}
$sharp$	ns_{frame}
$sharp_{\text{ref}}$	ns_{frame}
fps	ns_{frame}

OUTPUT:

Name	Dimension
q_{frame}	ns_{frame}
q_sec	$video\_duration\_in\_sec$
q	1
$O.22$	1
$O.27$	1

A weighting factor is computed by first computing $s_{\max}$ as the maximum over all orientations of s,

$$s_{\max}[n,i,j] = \max_k \quad s[n, dissim_{res}, k, i, j], \tag{40}$$

and then computing the weighting factor w_s as

$$w_s[n,i,j] = \left(\frac{1}{\max(0, c_{\text{lim}} - c_{\text{scale}} \cdot s_{\max}[n,i,j])}\right)^{c_{\text{exp}}} \tag{41}$$

and an additional border weighting is applied using

$$w[n,i,j] = \frac{d_{\text{border}}[i,j] + 1}{w_{max\_border\_dist}} \cdot w_s[n,i,j] \tag{42}$$

where d_{border} is the distance to the border, capped at $w_{max\_border\_dist} - 1$, i.e.,

$$d_{\text{border}}[i,j] = \min(i, j, (ns_h - 1 - i), (ns_w - 1 - j), (w_{max\_border\_dist}) - 1). \tag{43}$$

The weighting is applied to the dissimilarity values by

$$dissim_w[n] = dissim[n] \cdot \frac{w[n]}{E(w[n])} \tag{44}$$

$$dissim_{inc\_w}[n] = dissim_{inc}[n] \cdot \frac{w[n]}{E(w[n])}, \tag{45}$$

where the function E computes the average of $w[n,i,j]$ over all locators $i = 0..ns_h - 1, j = 0..ns_w - 1$.

A frame rate degradation is computed by

$$d_{fps} = (1 - S_{fps}(fps)) \cdot (1 - \exp(E(motion\_avg)/par_{motion\_fps})) \tag{46}$$

A motion weighting is computed by

$$mo_{avg\_weight} = 1 - par_{motion\_c} \cdot S_{mo}(motion\_avg) \quad (47)$$

Further, an additional correction is computed by

$$dissim_{cor} = L(dissim_{w}) \quad (48)$$

$$dissim_{inc\_cor} = L(dissim_{inc}), \quad (49)$$

where

$$L(s)[n,i,j] = s[n,i,j] \cdot (1 + par_{lum\_fac} * (1 + y_{lowres}[n,p,q])^{par_{lum\_exp}} \quad (50)$$

with $p = floor(i * y_{lowres_{height}}/ns_{h})$ and $q = floor(j * y_{lowres_{width}}/ns_{w})$.

Further, the values are mapped to range [0,1] by

$$d_{dis}[n,i,j] = mo_{avg\_weight}[n] \cdot S_{dis}(dissim_{corr}[n,i,j]) \quad (51)$$

$$d_{dis\_inc}[n,i,j] = mo_{avg\_weight}[n] \cdot S_{dis\_inc}(dissim_{inc\_corr}[n,i,j]) \quad (52)$$

A degradation due to loss in sharpness d_sharp is computed by

$$d_{sharp} = 1 - S_{rel\_sharp}\left(\min(1.0, \frac{sharp + c_{sharp}}{sharp\_ref + c_{sharp}})\right) \quad (53)$$

using the S-transformation S_{rel_sharp}, on the other hand, a degradation related to an increase in sharpness is computed by

$$d_{sharp\_inc} = S_{sharp\_inc}(\max(0.0, sharp - sharp\_ref)) \quad (54)$$

using the S-transformation S_{sharp_inc}. For a degradation factor, higher values are worse. For a quality factor, higher values are good. Degradation factors in the range $[0,1]$ can be inverted to quality factors by subtraction from one. Hence, a quality estimate is computed by a product of inverted degradations,

$$q_{\text{frame}_l} = \quad (55)$$

$$(1 - d_{sharp}) \cdot (1 - d_{sharp\_inc}) \cdot (1 - d_{fps}) \cdot E_{spat}((1 - d_{dis}) \cdot (1 - d_{dis\_inc})),$$

where E_{spat} computes the spatial mean, in more detail, for the sequence of matrices the mean for each sequence element is computed. To model temporal aspects of degradations the following is computed:

$$q_{\text{frame}} = 1 - deg\_fade\_out((1 - q\_frame_{l}), dt) \quad (56)$$

where the function deg_fade_out is described in the next paragraph using pseudo code.

For a vector $(v_i)_{i=0,..,N-1}$ and a monotonically increasing vector $(t_i)_{i=0,..,N}$ let $S_{v,t}$ denote a step function, a real-valued function taking the constant value v_i over the interval $[t_i, t_{i+1}]$. Let $\text{Avg}(S_{v,t}, t_0, t_1)$ denote the average value of the step function $S_{v,t}$ over the interval $[t_0, t_1]$. Then the function *deg_fade_out* is defined by the following pseudo code:

```
function deg_fade_out(v,t)

  w = [0,0,...,0]
  for i in [1,2,..]:
    a = exp(-par_fade_dt)
    v_avg = Avg(S_{v,t},t[i+1]-par_fade_smooth,t[i+1])
    w[i] = max(v_avg,a * w[i-1] + (1-a) * v_avg)

  return w
```

The values of the parameters $par\_fade\_dt, par\_fade\_smooth$ are specified in clauses 10.14 and 10.13, respectively..

The temporal average quality q for the entire sequence is given by

$$q = E(q_{\mathrm{frame}}) \tag{57}$$

Furthermore, a per-second score is computed using the step function resampling approach in clause 10.9, resulting in q_{sec}. These [0,1]-valued scores $q, q_{\mathrm{frame}}, q_{\mathrm{sec}}$ are mapped to the range [1,5] by the rescale function $r: x \mapsto 4x + 1$ resulting in

$$O.27 = r(q), \quad O.22 = r(q_{\mathrm{sec}}) \tag{58}$$

Higher values of $q, O.22, O.27$ mean higher quality.

10.12 Viewing distance

For a relative viewing distance above 4*H*, the parameters for the mobile case are used. For a relative viewing distance below 2*H*, the parameters for the PC/TV case are used. For intermediate viewing distances, the parameters are linearly interpolated between the two.

10.13 Parameters

Name	Value
f_width_init	1 920
f_height_init	1 080
y_range_max	255
y_rescale	20
$n_{resolution}$	4
n_h	18
n_w	32
num_{h_res}	2
c_{lat}	0.3
d_{lat}	0.001
Δ_{lat}	2
L_o	8
q_{pos}	2
sharp_scale_fac	10.0
$stat\_scale\_fac$	4*255
y_low_res_width	5
y_low_res_height	3
c_sharp	0.05
w_max_border_dist	3
dissim_res	1
fps_chunk_dur	2
n_orient	8
ns_h	7
ns_w	14
sharp_frac	0.05

num_a	3
num_s	200
par_weight_scale	100
par_fade_smooth	0.5

10.14 Device-dependent model parameters

Name	Value PC/TV	Value MO/TA
par_weight_lim	5.593268792046344	4.656208421713784
par_weight_exp	0.9985031497295792	0.9999821534030532
par_{motion_fps}	0.10338749688116727	0.1000006225291463
par_{motion_c}	0.9683245820065315	0.7604347879732595
par_{lum_fac}	0.5573475746950503	0.5574799921101337
par_{lum_exp}	0.10014977581205474	0.10412368985745854
par_fade_dt	0.1616170238997139	0.1871980057940932

10.15 *S*-transformation parameters

Transform PC/TV	*x*-Position	*y*-Position	Slope
S_{mo}	1.0464757777038356	0.5	0.47124514999456596
S_{dis}	0.5450173005392799	0.7980273056330967	2.048041212706822
S_{dis_inc}	0.36420555146972666	0.6165825542863502	2.235668875917247
S_{rel_sharp}	0.6745913663781392	0.5	2.177200231342128
S_{sharp_inc}	0.289504984526356	0.5	2.028729717455461
S_{fps}	15.0	0.7500024932923486	0.01805843377341594
Transform Mobile	*x*-Position	*y*-Position	Slope
S_{mo}	1.2972708989704074	0.5	0.1882251589297096
S_{dis}	0.7211019847289146	0.6830850971844077	2.3914975476194362
S_{dis_inc}	0.4041098766701082	0.5404927853257431	1.3109987046856608
S_{rel_sharp}	0.28071248315138375	0.5	0.9889249368712523
S_{sharp_inc}	0.6740897012131203	0.5	2.9946362074534
S_{fps}	15.0	0.7665500949169916	0.021999942089236887

Appendix I

Performance figures

(This appendix does not form an integral part of this Recommendation.)

In this appendix, RMSE is reported, for the model used stand-alone. Note that the numbers are reported after a final per-database mapping between the model output and the subjective scores of a database. This linear mapping is used to account for scale and bias variations between different databases.

Table I.1 – Validation performance of model used stand-alone: The *submitted model* is the model trained on the exchanged training databases and frozen before creation of validation data. The model was retrained using a fivefold cross-validation approach, with the validation performance listed to show stability of the performance indicating no over-fitting

Pixel-based reference	*Submitted model*	0.444				
	Five-fold cross-validation	0.418	0.418	0.425	0.442	0.429

The re-training of the submitted model was performed on five different splits. The splits were defined on the database level. The following is the procedure that was followed to determine the splits.

- All training and validation databases were merged to obtain in total 26 different short databases (18 PC/TV and 8 MO/TA).
- A level of difficulty of prediction for each database was determined based on average prediction error over all models.
- A 50-50 training-validation split was determined randomly but respecting the level of difficulty. In total, five different splits were defined. Each split had a balanced distribution of databases based on difficulty in both the training and validation.
- The 50-50 split was separately performed for PC/TV and MO/TA cases.
- The final model coefficients correspond to the best performing split.

Bibliography

[b-ITU-T G.1022] Recommendation ITU-T G.1022 (2016), *Buffer models for media streams on TCP transport.*

[b-ITU-T P.800.1] Recommendation ITU-T P.800.1 (2016), *Mean opinion score (MOS) terminology*.

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

[b-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*.

[b-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*.

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

[b-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*.

[b-ITU-T P.1203] Recommendation ITU-T P.1203 (2017), *Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport*.

[b-ITU-T P.1203.2] Recommendation ITU-T P.1203.2 (2017), *Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport – Audio quality estimation module.*

[b-ITU-T P.1204] Recommendation ITU-T P.1204 (2020), *Video quality assessment of streaming services over reliable transport for resolutions up to 4K.*

[b-ITU-T P.1204.3] Recommendation ITU-T P.1204.3 (2020), *Video quality assessment of streaming services over reliable transport for resolutions up to 4K with access to full bitstream information.*

[b-ITU-T P.1204.5] Recommendation ITU-T P.1204.5 (2020), *Video quality assessment of streaming services over reliable transport for resolutions up to 4K with access to transport and received pixel information.*

[b-ITU-T P.1401] Recommendation ITU-T P.1401 (2020), *Methods, metrics and procedures for statistical evaluation, qualification and comparison of objective quality prediction models*.

[b-FFmpeg] FFmpeg Developers (Internet). *FFmpeg: A complete, cross-platform solution to record, convert and stream audio and video,* version 4.2, "Ada" [Software]. Available [viewed 2020-02-26] from: http://ffmpeg.org/\

[b-Huber] Scikit-learn developers (2007-19). Huber regression. In: *Scikit-learn*, version 0.22.1 [Software]. Available [viewed 2020-02-26] from: https://scikit-learn.org/stable/modules/linear_model.html#huber-regression

[b-Scikit-learn] Scikit-learn developers (2007-19). *Scikit-learn*, version 0.22.1 [Software]. Available [viewed 2020-02-26] from: https://scikit-learn.org/stable/whats_new/v0.19.html#version-0-19-1

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, 2020