International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.247

(08/2008)

SERIES J: CABLE NETWORKS AND TRANSMISSION OF TELEVISION, SOUND PROGRAMME AND OTHER MULTIMEDIA SIGNALS

Measurement of the quality of service

Objective perceptual multimedia video quality measurement in the presence of a full reference

Recommendation ITU-T J.247

Recommendation ITU-T J.247

Objective perceptual multimedia video quality measurement in the presence of a full reference

Summary

The term multimedia, as defined in Recommendation ITU-T J.148, is the combination of multiple forms of media such as video, audio, text, graphics, fax and telephony in the communication of information. A three stage approach has been adopted to recommending objective assessment methods for multimedia. The first two stages identify perceptual quality tools appropriate for measuring video and audio individually. The third stage will identify objective assessment methods for the combined audiovisual media. This Recommendation contains the first stage – video only used in multimedia applications.

Recommendation ITU-T J.247 provides guidelines on the selection of appropriate objective perceptual video quality measurement methods when a full reference signal is available. The following are example applications that can use this Recommendation:

1) Internet multimedia streaming.
2) Video telephony and conferencing over cable and other networks.
3) Progressive video television streams viewed on LCD monitors over cable networks, including those transmitted over the Internet using Internet Protocol (VGA was the maximum resolution in the validation test).
4) Mobile video streaming over telecommunications networks.
5) Some forms of IPTV video payloads (VGA was the maximum resolution in this validation test).

Source

Recommendation ITU-T J.247 was approved on 13 August 2008 by ITU-T Study Group 9 (2005-2008) under Recommendation ITU-T A.8 procedures.

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/.

CONTENTS

Page

Recommendation ITU-T J.247

Objective perceptual multimedia video quality measurement in the presence of a full reference

1 Scope

This Recommendation provides guidelines and recommendations on the selection of appropriate perceptual video quality measurement equipment for use in multimedia applications when the full reference measurement method can be used.

The full reference measurement method can be used when the unimpaired reference video signal is readily available at the measurement point, as may be the case of measurements on individual equipment or a chain in the laboratory or in a closed environment such as a cable television head-end. The estimation methods are based on processing video in VGA, CIF and QCIF resolution.

The validation test material contained both multiple coding degradations and various transmission error conditions (e.g., bit errors, dropped packets). In the case where coding distortions are considered in the video signals, the encoder can utilize various compression methods (e.g., MPEG-2, ITU-T H.264, etc.). The models proposed in this Recommendation may be used to monitor the quality of deployed networks to ensure their operational readiness. The visual effects of the degradations may include spatial as well as temporal degradations (e.g., frame repeats, frame skips, frame rate reduction). The models in this Recommendation can also be used for lab testing of video systems. When used to compare different video systems, it is advisable to use a quantitative method (such as that in [ITU-T J.149]) to determine the models' accuracy for that particular context.

This Recommendation is deemed appropriate for telecommunications services delivered at 4 Mbit/s or less presented on mobile/PDA and computer desktop monitors. The following conditions were allowed in the validation test for each resolution:

- PDA/mobile (QCIF): 16 kbit/s to 320 kbit/s.
- CIF: 64 kbit/s-2 Mbit/s (C01 has several 2 Mbit/s).
- VGA: 128 kbit/s-4 Mbit/s (V13 has one HRC with 6 Mbit/s).

Table 1 – Factors for which this Recommendation has been evaluated

Test factors
Transmission errors with packet loss
Video resolution QCIF, CIF and VGA
Video bit rates: QCIF: 16 kbit/s to 320 kbit/s CIF: 64 kbit/s-2 Mbit/s VGA: 128 kbit/s-4 Mbit/s
Temporal errors (pausing with skipping) of maximum 2 seconds
Video frame rates from 5 fps to 30 fps
Coding technologies
H.264/AVC (MPEG-4 part 10), VC-1, Windows Media 9, Real Video (RV 10), MPEG-4 Part 2. (see Note below).

Table 1 – Factors for which this Recommendation has been evaluated

Applications
Real-time, in-service quality monitoring at the source
Remote destination quality monitoring when a copy of the source is available
Quality measurement for monitoring of a storage or transmission system that utilizes video compression and decompression techniques, either a single pass or a concatenation of such techniques
Lab testing of video systems
NOTE – The validation testing of models included video sequences encoded using 15 different video codecs. The five codecs listed above were most commonly applied to encode test sequences and any recommended models may be considered appropriate for evaluating these codecs. In addition to these five codecs, a smaller proportion of test sequences were created using the following codecs: Cinepak, DivX, ITU-T H.261, ITU-T H.263, ITU-T H.263+[1], JPEG-2000, MPEG-1, MPEG-2, Sorenson, ITU-T H.264 SVC, Theora. It can be noted that some of these codecs were used only for CIF and QCIF resolutions because they are expected to be used in the field mostly for these resolutions. Before applying a model to sequences encoded using one of these codecs, the user should carefully examine its predictive performance to determine whether the model reaches acceptable predictive performance.

1.1 Application

This Recommendation provides video quality estimations for video classes TV3 to MM5B, as defined in Annex B of [ITU-T P.911]. Note that the maximum resolution was VGA and the maximum bit rate covered well in the test was 4 Mbit/s. The applications for the estimation models described in this Recommendation include, but are not limited to:

1) potentially real-time, in-service quality monitoring at the source;
2) remote destination quality monitoring when a copy of the source is available;
3) quality measurement for monitoring of a storage or transmission system that utilizes video compression and decompression techniques, either a single pass or a concatenation of such techniques;
4) lab testing of video systems.

1.2 Model usage

All four models significantly outperform PSNR.

The models from OPTICOM and Psytechnics tend to perform slightly better than the NTT and Yonsei models in some resolutions. The Psytechnics and OPTICOM models usually produce statistically equivalent results. For QCIF, the model from NTT is often statistically equivalent to the models of Psytechnics and OPTICOM. For VGA, the Yonsei model is typically statistically equivalent to the OPTICOM and Psytechnics models. The tables below provide an overview on the models' performance.

Although all four models can be used to meet the industries' needs adequately, for VGA, it is strongly advised that the models from OPTICOM, Psytechnics or Yonsei be used to obtain slightly better performance in most cases. For the same reason, it is strongly advised that the models from OPTICOM or Psytechnics be used for CIF. For the same reason, it is strongly advised that the models from NTT, OPTICOM or Psytechnics be used for QCIF.

[1] ITU-T H.263+ is a particular configuration of ITU-T H.263.

The OPTICOM model shows the best overall minimum correlation. The minimum correlation coefficients of the OPTICOM model are 0.68, of the NTT model 0.60, of the Yonsei model 0.59, and of the Psytechnics model 0.57, respectively.

The Psytechnics model obtained the highest number of occurrences of being in the top group. The total number of occurrences in the top group is 37 for the Psytechnics, 34 for the OPTICOM, 25 for the NTT and 24 for the Yonsei model.

Model performance overview

VGA	NTT	OPTICOM	Psytechnics	Yonsei	PSNR
Avg. correlation	0.786	0.825	0.822	0.805	0.713
Min. correlation	0.598	0.685	0.565	0.612	0.499
Occurrences at rank 1	8	10	11	10	3
Ranking analysis	Second	Best	Best	Best	–

CIF	NTT	OPTICOM	Psytechnics	Yonsei	PSNR
Avg. correlation	0.777	0.808	0.836	0.785	0.656
Min. correlation	0.675	0.695	0.769	0.712	0.440
Occurrences at rank 1	8	13	14	10	0
Ranking analysis	Second	Best	Best	Second	–

QCIF	NTT	OPTICOM	Psytechnics	Yonsei	PSNR
Avg. correlation	0.819	0.841	0.830	0.756	0.662
Min. correlation	0.711	0.724	0.664	0.587	0.540
Occurrences at rank 1	9	11	12	4	1
Ranking analysis	Best	Best	Best	Second	–

1.3 Limitations

The estimation models described in this Recommendation cannot be used to fully replace subjective testing. Correlation values between two carefully designed and executed subjective tests (i.e., in two different laboratories) normally fall within the range 0.95 to 0.98. If this Recommendation is utilized to make video system comparisons (e.g., comparing two codecs), it is advisable to use a quantitative method (such as that in [ITU-T J.149]) to determine the models' accuracy for that particular context.

The models in this Recommendation were validated by measuring video that exhibits frame freezes up to 2 seconds.

The models in this Recommendation were not validated for measuring video that has a steadily increasing delay (e.g., video which does not discard missing frames after a frame freeze).

It should be noted that in case of new coding and transmission technologies producing artefacts which were not included in this evaluation, the objective models may produce erroneous results. Here a subjective evaluation is required.

2 References

2.1 Normative 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 J.143] Recommendation ITU-T J.143 (2000), *User requirements for objective perceptual video quality measurements in digital cable television*.

[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*.

2.2 Informative references

[ITU-T H.261] Recommendation ITU-T H.261 (1993), *Video codec for audiovisual services at p x 64 kbit/s*.

[ITU-T H.263] Recommendation ITU-T H.263 (1996), *Video coding for low bit rate communication*.

[ITU-T H.263] Recommendation ITU-T H.263 (1998), *Video coding for low bit rate communication.* (ITU-T H.263+)

[ITU-T H.264] Recommendation ITU-T H.264 (2003), *Advanced video coding for generic audiovisual services*.

[ITU-T J.144] Recommendation ITU-T J.144 (2001), *Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference.*

[ITU-T J.148] Recommendation ITU-T J.148 (2003), *Requirements for an objective perceptual multimedia quality model*.

[ITU-T J.149] Recommendation ITU-T J.149 (2004), *Method for specifying accuracy and cross-calibration of Video Quality Metrics (VQM)*.

[ITU-T J.244] Recommendation ITU-T J.244 (2008), *Full reference and reduced reference calibration methods for video transmission systems with constant misalignment of spatial and temporal domains with constant gain and offset*.

[ITU-T P.931] Recommendation ITU-T P.931 (1998), *Multimedia communications delay, synchronization and frame rate measurement*.

[ITU-R BT.500-11] Recommendation ITU-R BT.500-11 (2002), *Methodology for the subjective assessment of the quality of television pictures*.

[VQEG] *Final report from the Video Quality Experts Group on the validation of objective models of multimedia quality assessment, Phase I 2008*.
<ftp://vqeg.its.bldrdoc.gov/Documents/Projects/multimedia/MM_Final_Report/VQEG_MM_Report_Final_v2.6.pdf>

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 objective perceptual measurement (picture): [ITU-T J.144].

3.1.2 proponent: [ITU-T J.144].

3.1.3 subjective assessment (picture): [ITU-T J.144].

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 anomalous frame repetition: An event where the HRC outputs a single frame repeatedly in response to an unusual or out of the ordinary event. Anomalous frame repetition includes, but is not limited to, the following types of events: an error in the transmission channel, a change in the delay through the transmission channel, limited computer resources impacting the decoder's performance and limited computer resources impacting the display of the video signal.

3.2.2 constant frame skipping: An event where the HRC outputs frames with updated content at an effective frame rate that is fixed and less than the source frame rate.

3.2.3 effective frame rate: The number of unique frames (i.e., total frames – repeated frames) per second.

3.2.4 frame rate: The number of (progressive) frames displayed per second (fps).

3.2.5 intended frame rate: The number of video frames per second physically stored for some representation of a video sequence. The intended frame rate may be constant or may change with time. Two examples of constant intended frame rates are a BetacamSP tape containing 25 fps and a VQEG FR-TV phase I compliant 625-line YUV file containing 25 fps; these both have an intended frame rate of 25 fps. One example of a variable intended frame rate is a computer file containing only new frames; in this case, the intended frame rate exactly matches the effective frame rate. The content of video frames is not considered when determining intended frame rate.

3.2.6 live network conditions: Errors imposed upon the digital video bit stream as a result of live network conditions. Examples of error sources include packet loss due to heavy network traffic, increased delay due to transmission route changes, multi-path on a broadcast signal and fingerprints on a DVD. Live network conditions tend to be unpredictable and unrepeatable.

3.2.7 pausing with skipping: Events where the video pauses for some period of time and then restarts with some loss of video information. In pausing with skipping, the temporal delay through the system will vary about an average system delay, sometimes increasing and sometimes decreasing. One example of pausing with skipping is a pair of IP videophones, where heavy network traffic causes the IP videophone display to freeze briefly; when the IP videophone display continues, some content has been lost. Another example is a videoconferencing system that performs constant frame skipping or variable frame skipping. Constant frame skipping and variable frame skipping are subsets of pausing with skipping. A processed video sequence containing pausing with skipping will be approximately the same duration as the associated original video sequence.

3.2.8 pausing without skipping: Any event where the video pauses for some period of time and then restarts without losing any video information. Hence, the temporal delay through the system must increase. One example of pausing without skipping is a computer simultaneously downloading and playing an AVI file, where heavy network traffic causes the player to pause briefly and then continue playing. A processed video sequence containing pausing without skipping events will always be longer in duration than the associated original video sequence.

3.2.9 refresh rate: The rate at which the computer monitor is updated.

3.2.10 simulated transmission errors: Errors imposed upon the digital video bit stream in a highly controlled environment. Examples include simulated packet loss rates and simulated bit errors. Parameters used to control simulated transmission errors are well defined.

3.2.11 source frame rate (SFR): The intended frame rate of the original source video sequences. The source frame rate is constant. For the VQEG MM phase I test, the SFR was either 25 fps or 30 fps.

3.2.12 transmission errors: Any error imposed on the video transmission. Example types of errors include simulated transmission errors and live network conditions.

3.2.13 variable frame skipping: An event where the HRC outputs frames with updated content at an effective frame rate that changes with time. The temporal delay through the system will increase and decrease with time, varying about an average system delay. A processed video sequence containing variable frame skipping will be approximately the same duration as the associated original video sequence.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ACR	Absolute Category Rating (see [ITU-T P.910])
ACR-HR	Absolute Category Rating with Hidden Reference (see [ITU-T P.910])
AVI	Audio Video Interleave
CIF	Common Intermediate Format (352 x 288 pixels)
DMOS	Difference Mean Opinion Score
FR	Full Reference
FRTV	Full Reference Television
HRC	Hypothetical Reference Circuit
ILG	VQEG's Independent Laboratory Group
LCD	Liquid Crystal Display
MM	Multimedia
MOS	Mean Opinion Score
MOSp	Mean Opinion Score, predicted
NR	No (or zero) Reference
PDA	Personal Digital Assistant
PSNR	Peak Signal-to-Noise Ratio
PVS	Processed Video Sequence
QCIF	Quarter Common Intermediate Format (176 x 144 pixels)
RMSE	Root Mean Square Error
RR	Reduced Reference
SFR	Source Frame Rate
SRC	Source Reference Channel or Circuit
VGA	Video Graphics Array (640 x 480 pixels)
VQEG	Video Quality Experts Group
YUV	Color Space and file format

5 Conventions

None.

6 Description of the full reference measurement method

The double-ended measurement method with full reference, for objective measurement of perceptual video quality, evaluates the performance of systems by making a comparison between the undistorted input, or reference, video signal at the input of the system, and the degraded signal at the output of the system (Figure 1).

Figure 1 shows an example of application of the full reference method to test a codec in the laboratory.

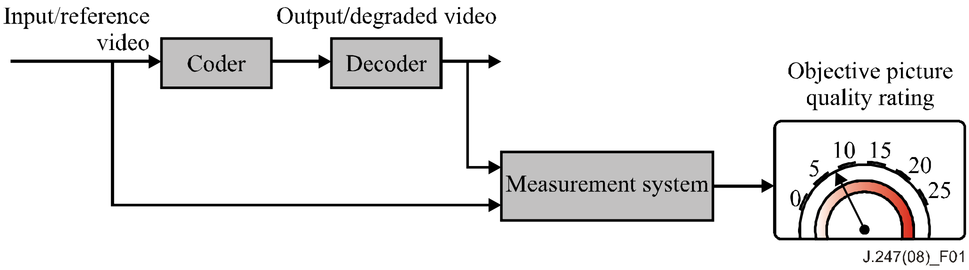


Figure 1 – Application of the full reference perceptual quality measurement method to test a codec in the laboratory

The comparison between input and output signals may require a temporal alignment or a spatial alignment process, the latter to compensate for any vertical or horizontal picture shifts or cropping. It also may require correction for any offsets or gain differences in both the luminance and the chrominance channels. The objective picture quality rating is then calculated, typically by applying a perceptual model of human vision.

Alignment and gain adjustment is known as registration. This process is required because most full reference methods compare reference and processed pictures on what is effectively a pixel-by-pixel basis. The video quality metrics described in Annexes A through D include registration methods.

As the video quality metrics are typically based on approximations to human visual responses, rather than on the measurement of specific coding artefacts, they are in principle equally valid for analogue systems and for digital systems. They are also in principle valid for chains where analogue and digital systems are mixed, or where digital compression systems are concatenated.

Figure 2 shows an example of the application of the full reference method to test a transmission chain.

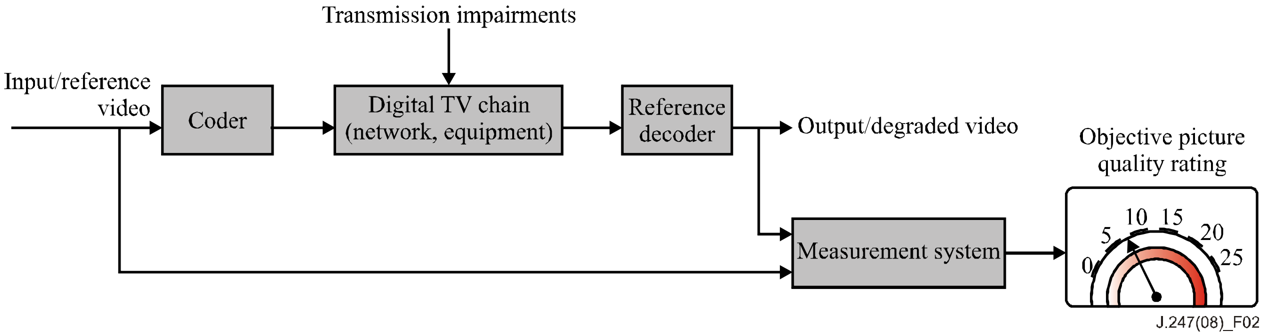


Figure 2 – Application of the full reference perceptual quality measurement method to test a transmission chain

In this case, a reference decoder is fed from various points in the transmission chain, e.g., the decoder can be located at a point in the network, as in Figure 2, or directly at the output of the encoder, as in Figure 1. If the digital transmission chain is transparent, the measurement of objective picture quality rating at the source is equal to the measurement at any subsequent point in the chain.

It is generally accepted that the full reference method provides the best accuracy for perceptual picture quality measurements. The method has been proven to have the potential for high correlation with subjective assessments made in conformity with the ACR-HR methods specified in [ITU-T P.910].

7 Findings of the Video Quality Experts Group (VQEG)

Studies of perceptual video quality measurements are conducted in an informal group, called the Video Quality Experts Group (VQEG), which reports to ITU-T Study Groups 9 and 12 and ITU-R Study Group 6. The recently completed multimedia phase I test of VQEG assessed the performance of proposed full reference perceptual video quality measurement algorithms for QCIF, CIF and VGA formats.

Based on present evidence, four methods can be recommended by ITU-T at this time. These are:

Annex A − VQEG Proponent A: NTT, Japan.

Annex B − VQEG Proponent B: OPTICOM, Germany.

Annex C − VQEG Proponent C: Psytechnics, UK.

Annex D − VQEG Proponent D: Yonsei University, Korea.

The technical descriptions of these models can be found in Annexes A through D, respectively. Note that the ordering of annexes is purely arbitrary and provides no indication of quality prediction performance.

Tables 2 to 4 below provide informative details on the models' performances in the VQEG multimedia phase I test.

Table 2 – VGA resolution: Informative description on the models' performances in the VQEG multimedia phase I test: Averages over 14 subjective tests

Metric	NTT	OPTICOM	Psytechnics	Yonsei	PSNR[2]
Annex	A	B	C	D	
Pearson correlation	0.786	0.825	0.822	0.805	0.713
RMS error	0.621	0.571	0.566	0.593	0.714
Outlier ratio	0.523	0.502	0.524	0.542	0.615

[2] The PSNR values reported here are taken from the VQEG multimedia phase I final report. These values were calculated by NTIA/ITS.

Table 3 – CIF resolution: Informative description on the models' performances in the VQEG multimedia phase I test: Averages over 14 subjective tests

Metric	NTT	OPTICOM	Psytechnics	Yonsei	PSNR[2]
Annex	A	B	C	D	
Pearson correlation	0.777	0.808	0.836	0.785	0.656
RMS error	0.604	0.562	0.526	0.594	0.720
Outlier ratio	0.538	0.513	0.507	0.522	0.632

Table 4 – QCIF resolution: Informative description on the models' performances in the VQEG multimedia phase I test: Averages over 14 subjective tests

Metric	NTT	OPTICOM	Psytechnics	Yonsei	PSNR[2]
Annex	A	B	C	D	
Pearson correlation	0.819	0. 841	0. 830	0. 756	0.662
RMS error	0.551	0.516	0.517	0.617	0.721
Outlier ratio	0.497	0.461	0.458	0.523	0.596

Based on each metric, each FR VGA model was in the group of top performing models the following number of times:

Statistic	NTT	OPTICOM	Psytechnics	Yonsei	PSNR
Correlation	8	10	11	10	3
RMSE	4	8	10	6	0
Outlier ratio	9	11	12	8	4

Based on each metric, each FR CIF model was in the group of top performing models the following number of times:

Statistic	NTT	OPTICOM	Psytechnics	Yonsei	PSNR
Correlation	8	13	14	10	0
RMSE	6	10	13	9	0
Outlier ratio	10	13	12	11	1

Based on each metric, each FR QCIF model was in the group of top performing models the following number of times:

Statistic	NTT	OPTICOM	Psytechnics	Yonsei	PSNR
Correlation	9	11	12	4	1
RMSE	7	10	11	2	1
Outlier ratio	10	11	12	8	4

NOTE – As a general guideline, small differences in these totals do not indicate an overall difference in performance.

Secondary analysis

The secondary analysis averages over all video sequences associated with each video system (or condition), and thus reflects how well the model tracks the average hypothetical reference circuit (HRC) performance. The following tables show the average correlations for each model and resolution in the secondary analysis.

VGA correlation

	NTT	OPTICOM	Psytechnics	Yonsei	PSNR
Average	0.891	0.914	0.903	0.864	0.809

CIF correlation

	NTT	OPTICOM	Psytechnics	Yonsei	PSNR
Average	0.915	0.919	0.913	0.892	0.817

QCIF correlation

	NTT	OPTICOM	Psytechnics	Yonsei	PSNR
Average	0.942	0.937	0.920	0.893	0.882

Annex A

NTT full reference method

(This annex forms an integral part of this Recommendation)

A.1 Introduction

The NTT model accurately estimates subjective video quality by a precise alignment process and a video quality algorithm that reflects human visual characteristics in consideration of the influence of codecs, bit rate, frame rate and video quality distorted by packet loss.

A.2 Model overview

The NTT model is divided into three software modules: a video alignment module, temporal/spatial feature amount derivation module, and subjective video quality estimation module (Figure A.1).

The video alignment module is divided into a macro alignment process and a micro alignment process. The macro alignment process matches pixels between reference video signals (RI) and processed video signals (PI) in spatial-temporal directions and filters the video sequences in consideration of the influence of video capturing and post-processing of the decoder. The micro alignment process matches frames between reference and processed video sequences in consideration of the influence of video frame skipping and freezing after the macro alignment process has finished.

The temporal/spatial feature amount derivation module calculates a spatial degradation parameter and a temporal degradation parameters (PC) by using an aligned reference video signal (RI') and an aligned processed video signal (PI'). The spatial degradation parameter is based on four parameters that reflect either the presence of overall noise, spurious edges, localized motion distortion or localized spatial distortion. The temporal degradation parameter, calculated by weighted freeze-length summation, reflects frame freezing and frame-rate variation.

The subjective video quality estimation module calculates the objective video quality (Q) by using the previously mentioned parameters. The details of the processes in each block are explained in the following clauses.

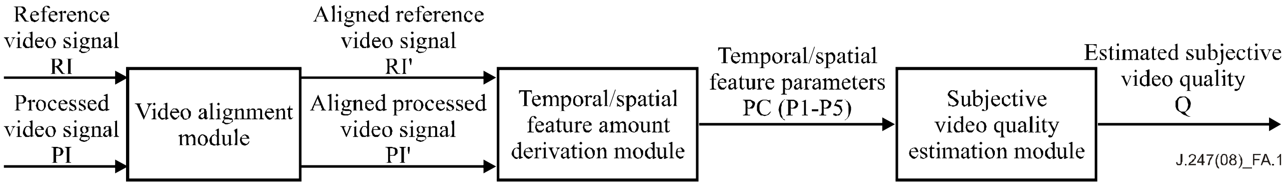

Reference video signal RI
Processed video signal PI
Video alignment module
Aligned reference video signal RI'
Aligned processed video signal PI'
Temporal/spatial feature amount derivation module
Temporal/spatial feature parameters PC (P1-P5)
Subjective video quality estimation module
Estimated subjective video quality Q
J.247(08)_FA.1

Figure A.1 – Block diagram of NTT model

A.3 Video alignment module

If every pair of pixels in the reference and processed video signals is aligned correctly, a full reference objective video assessment method cannot properly estimate subjective video quality. Therefore, the signals must be aligned before the evaluation process. In this alignment module, each frame of the processed video signal is associated with a frame of the reference video signal by comparing only luminance.

The macro alignment and micro alignment processes in the alignment module are shown in Figure A.2. The macro alignment process consists of i) alignment in spatial and temporal directions, ii) noise removal, and iii) alignment of gain/offset. These are performed once for each video sequence. Then, micro alignment is performed locally (i.e., frame by frame with a search window of several seconds) before calculating the objective video quality. The details of the process in the video alignment module are explained step by step. In all the alignment in this clause, invalid pixel data is excluded (e.g., pixels in the processed signal that do not have associated reference pixels because of spatial shift of the system under test).

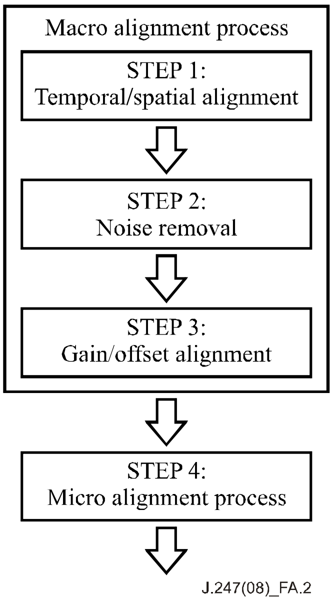


Figure A.2 – Process in video alignment module

Step 1: Temporal/spatial alignment process in one sequence

This process is performed once per pair of video clips, which are the reference and processed videos, to align them globally. For stable and accurate alignment, this process should be carried out using an error-free or non-anomalous error period (e.g., 1 second) of video signals. In the temporal/spatial alignment process, a processed video frame is moved up, down, left and right by pixels relative to a reference video frame and multiple frames before and after that reference frame, as shown in Figure A.3, to find the pixel and frame shifts that give the minimum difference in terms of luminance between the reference and processed videos. The search ranges in the spatial and temporal domains depend on the amount of shift expected in the system under test. Finally, all the pixels in the processed videos are shifted in the above directions.

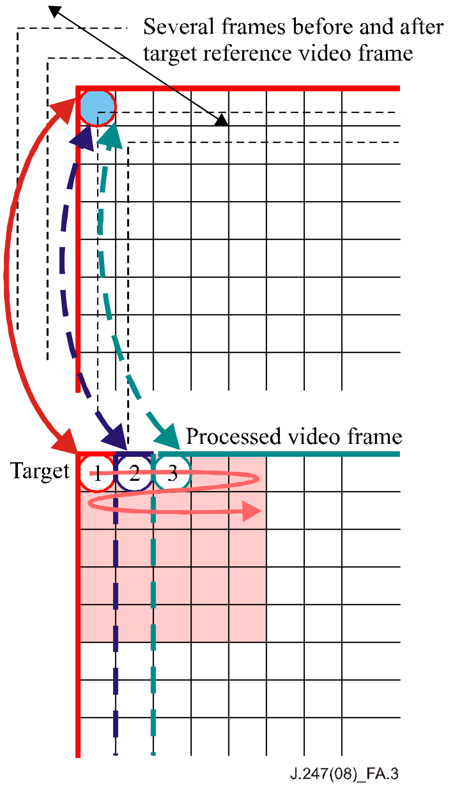


Figure A.3 – Temporal and spatial alignment process

Step 2: Noise removal process

The noise removal process removes the influence of noise on the processed video sequence. Noise is assumed to occur in cables or in the post-processing conducted in the decoder/player and video board. The following equation is applied for both the reference video and processed video signals.

$$Y(m,n) = \sum_{i=-k}^{i=k} \sum_{j=-l}^{j=l} X(m+i,n+j)W(i,j) \qquad \text{(A-1)}$$

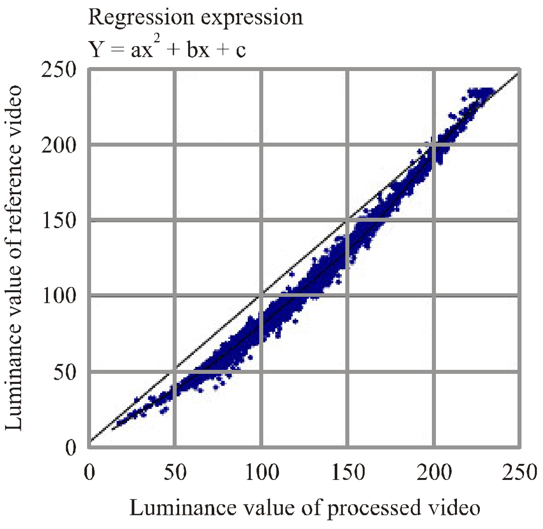
Here, $X(m,n)$ is a pixel of a video frame, and $W(i,j)$ is the cross-shaped median filter that is shown in Figure A.4.

0	1	0
1	1	1
0	1	0

Figure A.4 – Cross-shaped median filter

Step 3: Gain/offset alignment process

The gain/offset alignment process removes the bias (influence) due to colour arrangement in a decoder or a player (including a video board). In comparison with a reference video, the luminance value of the processed video signal is corrected by a quadratic approximation equation that represents the relationship between the reference video signal and processed video signal. A scatter plot of the luminance values of the processed and reference videos is shown in Figure A.5. For such a relationship, a regression equation ($Y = ax^2 + bx + c$) is derived. This process should also be performed using an error-free or non-anomalous error period (e.g., 1 second) of the video signals. After that, all of the processed video is converted to values to match the reference video by using this equation.



J.247(08)_FA.5

Figure A.5 – Post filter influence

Step 4: Micro alignment process

The micro alignment process is needed in addition to the macro alignment process since the frame shift between reference and processed video may vary due to frame-rate variation caused by, for example, packet loss. This process searches the associated reference video frame for each target frame in the processed video using a search window, the duration of which depends on the amount of frame shift expected in the system under test.

First, this process calculates the difference in terms of luminance between the target frame and the previous frame only using the processed video and evaluates whether the target frame is a repetition of the previous frame (i.e., freezing). Note that the state of the reference video is not taken into account.

Depending on the result of the above evaluation, the next step is one of the following. Here, "freezeCounter", whose initial value is set to 0, indicates the number of successive frozen frames until the current frame:

i) Both the target frame and previous one are not frozen frames, or the target frame is the first frame (freezeCounter = 0).

   To determine the reference video frame associated with the target frame in the processed video, this step searches the reference video frame that gives the minimum difference in terms of luminance from the target frame. The search starts from 0.25 s backward and ends at 2 s forward with respect to the reference frame that is associated with the previous target frame. In the case of the first frame analysis, this pivot of search is determined by the global frame shift calculated in the macro alignment process.

ii) The target frame is not a frozen frame but the previous one is (freezeCounter > 0).

   The search starts from 0.25 s backward and ends at 2 s forward with respect to the reference frame which is "freezeCounter" frames ahead of the frame that is associated with the previous target frame. Then, the freezeCounter is reset (freezeCounter == 0).

iii) The target frame is a frozen frame.

Since the target frame is the same as the previous one, the reference frame associated with the processed video does not change. The freezeCounter is incremented by one.

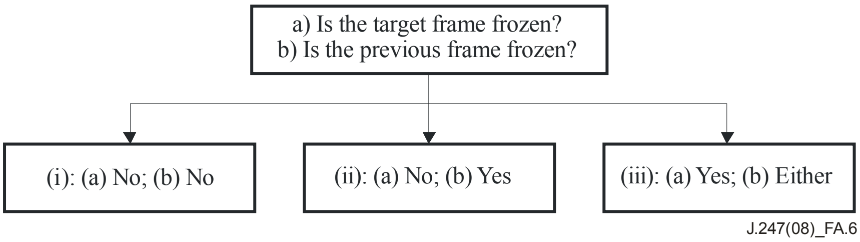

a) Is the target frame frozen?
b) Is the previous frame frozen?
(i): (a) No; (b) No
(ii): (a) No; (b) Yes
(iii): (a) Yes; (b) Either
J.247(08)_FA.6

Figure A.6 – Flow chart of conditions

A.4 Temporal/spatial feature amount derivation module

The temporal/spatial feature amount derivation module receives the aligned reference video signal and processed video signal that passed through the video alignment module and calculates the temporal/spatial feature (P1-P5) that is needed at the subjective video quality estimation module. The details of each parameter are explained in this clause. In addition, all the parameters are calculated only from luminance information. In all the analysis in this clause, invalid pixel data is excluded (e.g., pixels in the processed signal that do not have associated reference pixels because of spatial shift of the system under test).

A.4.1 P1: Peak signal-to-noise ratio (PSNR)

The peak signal-to-noise ratio (PSNR) reflects the overall degradation of processed video sequences (PVSs):

$$PSNR = \underset{m}{\mathrm{Ave}} PSNR(m) = \frac{1}{M}\sum_{m=0}^{M-1} 10\log_{10}\left(\frac{255^2}{MSE(m)}\right) \qquad \text{(A-2)}$$

where *M* is the total number of frames of the processed video and *MSE*(*m*) is the difference between the reference video and degraded video in the *m*-th frame, which is derived as:

$$MSE(m) = \frac{1}{N}\sum_{i,j}(Y_{out}(i,j,m) - Y_{in}(i,j,m))^2 \qquad \text{(A-3)}$$

The subscripts "*in*" and "*out*" indicate input and output, *N* is the total number of pixels and Y*(i, j, m)* is the luminance value (0-255) at pixel position *(i, j)* in the *m*-th frame. Note, however, that the maximum value of *PSNR* is truncated to 50.

A.4.2 P2: Logarithmic minimum HV- to non-HV-edge energy difference (*Log (-Min_HV)*)

This parameter specifically takes into account distortion in the form of block distortion by calculating the ratio between horizontal and vertical (*HV*) edges and other edges. Specifically, we derive this parameter by using equation A-4 to obtain the ratio between the total *SI* values (*SI* values for all pixels) for *HV* edges (i.e., *HV* edges that satisfy the conditions defined by the shaded regions of Figure A.7) and for other edges ($\overline{HV}$). r_{min} and Δθ are 20 and 0.05236, respectively.

$$\log(-(Min\_HV)) = \log_{10}\{-(\min_{m}\left\{\frac{HVR_{in}(m) - HVR_{out}(m)}{HVR_{in}(m)}\right\})\} \qquad \text{(A-4)}$$

If *Min_HV* is less than or equal to –1, *Min_HV* is forced to 0. The subscripts "*in*" and "*out*" again indicate input and output, and HVR is derived as follows:

$$HVR\ (\mathrm{m}) = \frac{HV(r_{min},\Delta\theta,m)+0.5}{\overline{HV}(r_{min},\Delta\theta,m)+0.5} \qquad \text{(A-5)}$$

The derivations of *HV* and $\overline{HV}$ are given below:

$$HV(r_{min},\Delta\theta,m) = \frac{1}{P}\sum_{i,j} SI_r(i,j,m) \qquad \text{(A-6)}$$

given that $SI_r(i,j,m) \geq r_{min} > 0$ and

$$k\frac{\pi}{2} - \Delta\theta < \tan^{-1}\left(\frac{SI_v(i,j,m\ )}{SI_h(i,j,m\ )}\right) < k\frac{\pi}{2} + \Delta\theta\ \ (k = 0, 1, 2, 3)$$

$$\overline{HV}(r_{min},\Delta\theta,m) = \frac{1}{P}\sum_{i,j} SI_r(i,j,m) \qquad \text{(A-7)}$$

given that $SI_r(i,j,m) \geq r_{min} > 0$ and

$$k\frac{\pi}{2} + \Delta\theta < \tan^{-1}\left(\frac{SI_v(i,j,m)}{SI_h(i,j,m)}\right) \leq (k+1)\frac{\pi}{2} - \Delta\theta\ \ (k = 0, 1, 2, 3)$$

SI(i,j,m) is derived as follows:

$$SI(i,j,m)\ = \max_{m}\left\{\frac{1}{N}\sum_{i,j}\{SI_h^2(i,j,m) + SI_v^2(i,j,m)\} - \left(\frac{1}{\mathrm{N}}\sqrt{SI_h^2(i,j,m) + SI_v^2(i,j,m)}\right)^2\right\}^{1/2} \qquad \text{(A-8)}$$

$$\begin{aligned} SI_h(i, j, m) = \{&-Y(i-1, j-1, m) + Y(i+1, j-1, m) \\ &-Y(i-1, j, m) + 2Y(i+1, j, m) \\ &-Y(i-1, j+1, m) + Y(i+1, j+1, m)\}, \end{aligned}$$

$$\begin{aligned} SI_v(i, j, m) = \{&-Y(i-1, j-1, m) - Y(i, j-1, m) \\ &-Y(i+1, j-1, m) + Y(i-1, j+1, m) \\ &+2Y(i, j+1, m) + Y(i+1, j+1, m)\}. \end{aligned}$$

$Y(i, j, m)$ is the luminance value (0-255) at pixel position (i, j) in the m-th frame, and SI_h and SI_v are pixel-by-pixel values for edge content in the horizontal and vertical directions, respectively.

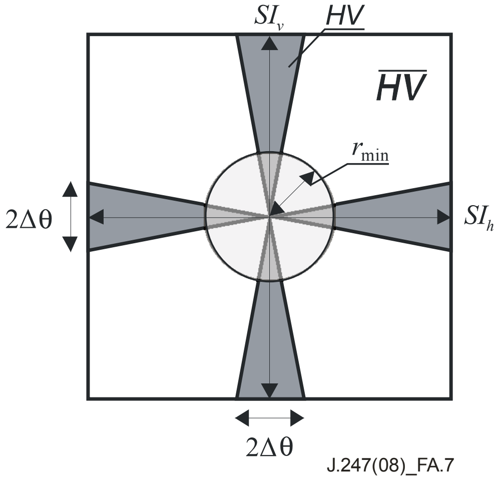


Figure A.7 – Graphical representation of HV features (shaded regions)

A.4.3 P3: Average moving energy of blocks (*Ave_MEB*)

The per-block average moving energy specifically represents motion distortion and localized distortion, which are underestimated or not represented by *PSNR*. To calculate this, the frame-to-frame differences are expressed in terms of luminance for each 8x8-pixel block. This is performed for both reference and processed videos (equation A-9). Then, *Ave_MEB* is calculated by using equations A-10 and A-11.

$$TI_b(k,l,m) = \frac{1}{64}\sum_{(i,j)}\{Y(8\times k+i,8\times l+j,m) - Y(8\times k+i,8\times l+j,m-1)\}^2 \qquad \text{(A-9)}$$

$$MEB(k,l,m) = \frac{TI_{b\_in}(k,l,m) - TI_{b\_out}(k,l,m)}{TI_{b\_in}(k,l,m)} \qquad \text{(A-10)}$$

where subscripts "*in*" and "*out*" indicate reference video and distorted video, respectively. *TI*$_b$ is the difference per 8x8-pixel block between before and after frames.

$$Ave\_MEB = \frac{1}{M}\sum_{m=0}^{M-1}\frac{1}{N_b}\sqrt{\sum_{(k,l)}\left(MEB(k,l,m)\right)^2} \qquad \text{(A-11)}$$

where N_b is the number of blocks.

A.4.4 P4: Fluctuating variance of local moving energy (*FV_LME*)

This parameter reflects temporal variance of partial spatial distortion and expresses the degree of variation of partial distortion in the sequence. First, *TI*$_b$ is derived by using equation A-9 (see Figure A-8), and the difference between reference and processed videos is calculated (equation A-12). Then, only for blocks that give the uppermost 10% of these values, the standard deviation of *MEB(k,l,m)* is calculated (see Figure A.9). Finally, the standard deviation of these values for all the frames is determined as *FV_LME*.

$$FV\_LME = \underset{m=0}{\overset{M-1}{Stdev}}\left\{\sum_{m=0}^{M-1}\underset{(k,l)}{\overset{\max 10\%}{Stdev}}\{MEB(k,l,m)\}\right\} \qquad \text{(A-12)}$$

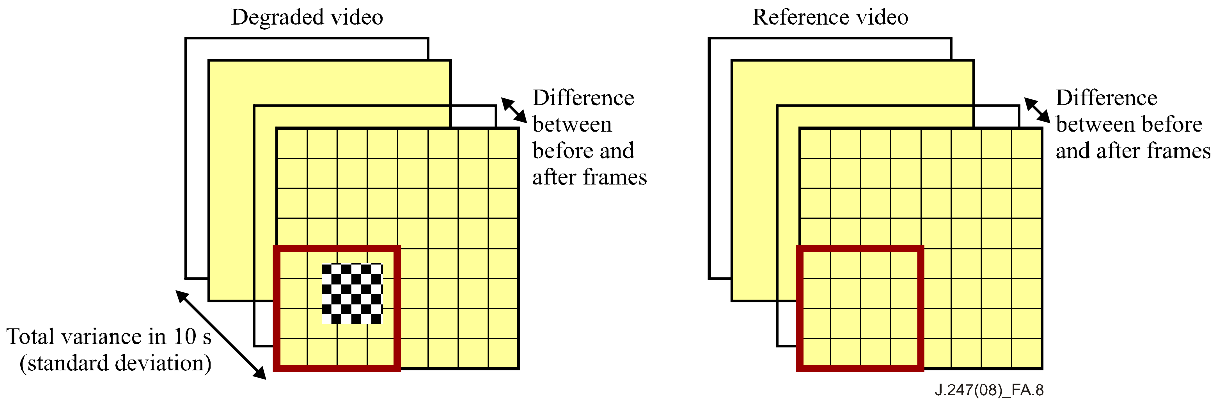

Degraded video
Difference between before and after frames
Total variance in 10 s (standard deviation)
Reference video
Difference between before and after frames
J.247(08)_FA.8

Figure A.8 – Variance of localized spatial distortion

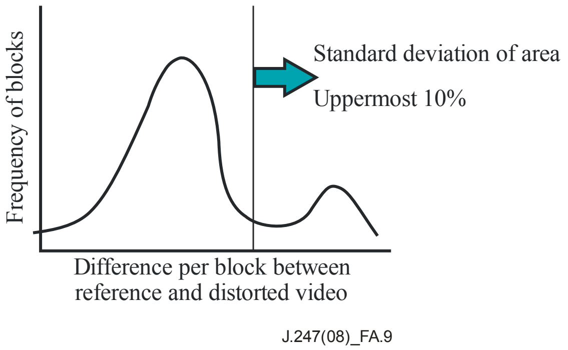


Figure A.9 – Area of localized spatial distortion

A.4.5 P5: Equivalent freeze length (*EFL*)

This parameter is the value that expresses the characteristics of freeze and frame rate reduction, which is derived by using only the distorted video signal from the alignment process. This process derives the freeze parameter by aggregating two freeze distortions into one equivalent freeze distortion, whose lengths are $EFL_1, \cdots, EFL_{n-1}$ in Figure A.10, while preserving the impact against subjective video quality, and repeating this process recursively for all freeze distortions in the assessment time. Finally, one can obtain an equivalent freeze length $EFL\left(= EFL_n\right)$ (Figure A.10), which expresses the subjective effects of all freeze distortions in the assessment time.

A detailed flow of the above process is shown in Figure A.10. First, when the first freeze distortion (freeze 1) occurs, FL_1, which is the length of freeze 1, is regarded as the equivalent freeze length EFL_1. Next, when a second freeze distortion (freeze 2) occurs, the equivalent freeze length EFL_2 is derived by aggregating the equivalent freeze length EFL_1, which expresses the impact of the first freeze distortion on subjective video quality, and the second freeze length FL_2 by using integration function *f*. After that, when a third freeze distortion (freeze 3) occurs, the equivalent freeze length EFL_3 is derived by aggregating the equivalent freeze length EFL_2, which expresses the impact of the first and second freeze distortions on subjective video quality, and the third freeze length FL_3 by using integration function *f*. This aggregating process is recursively executed for all freeze distortions.

The above process is summarized as follows. When a *k*-th freeze distortion (freeze *k*) occurs, the equivalent freeze length EFL_k is derived by aggregating the equivalent freeze length EFL_{k-1}, which expresses the impact of the first to (*k*-1)-th freeze distortion on subjective video quality, and *k*-th freeze length FL_k by using integration function *f*. This process is executed recursively in the range of $(2 \le k \le n)$, and the resultant equivalent freeze length EFL_n is the freeze parameter (EFL).

The integration function *f* is expressed as follows by using the new freeze length FL_k :

i) If $FL_k > 8$,

$$\begin{cases} EFL_k = EFL_{k-1} + FL_k \\ EFL_1 = FL_1 \\ 2 \le k \le n \end{cases}$$

ii) If $FL_k \le 8$,

$$\begin{cases} x_{k-1} = g_i^{-1}(EFL_{k-1}) \\ x_k = x_{k-1} + FL_k \\ EFL_k = g_i(x_k) \\ EFL_1 = FL_1 \\ 2 \le k \le n \end{cases}$$

NOTE – $g_i(x) = p_i x + q_i \log(x) + r_i$, g_i^{-1} is the inverse function of g_i, and $i = FL_k$. Examples of coefficients p_i, q_i and r_i are shown in Table A.1.

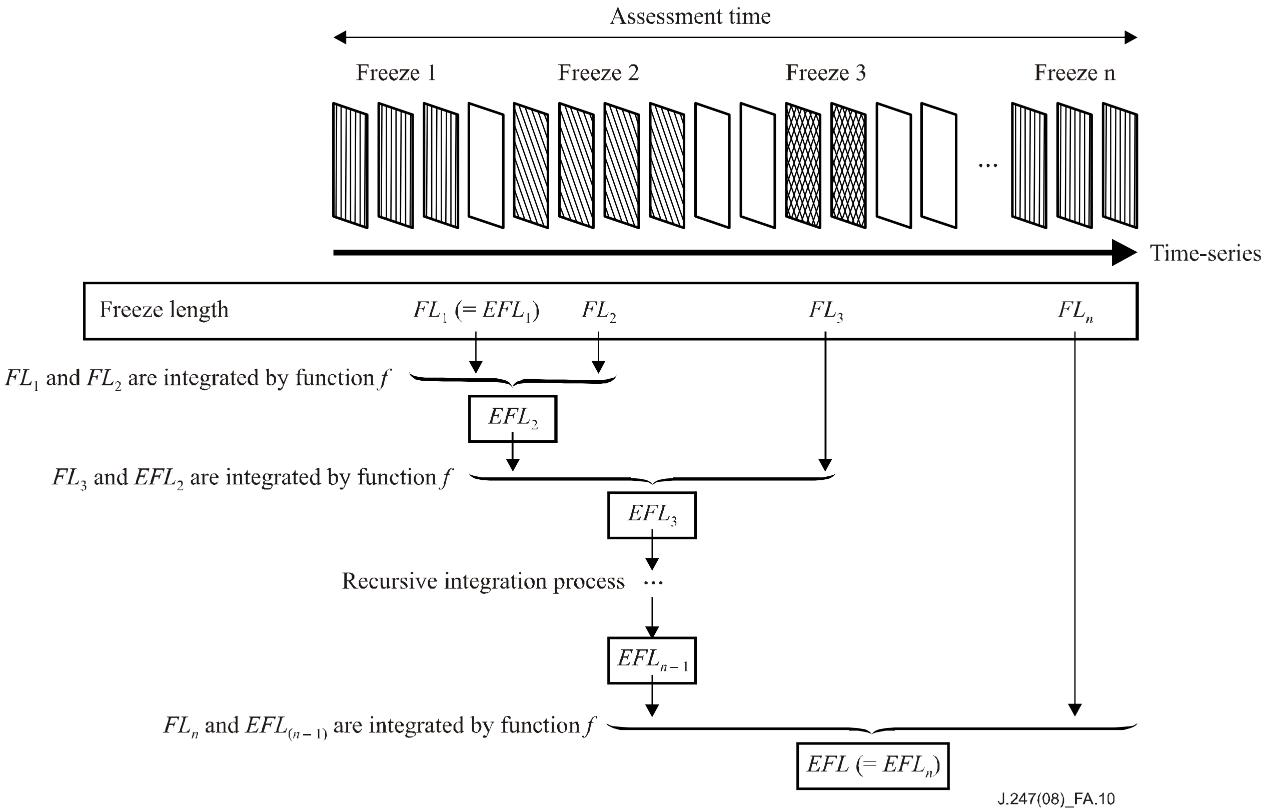

Assessment time
Freeze 1
Freeze 2
Freeze 3
Freeze n
...
Time-series
Freeze length
FL_1 (= EFL_1)
FL_2
FL_3
FL_n
FL_1 and FL_2 are integrated by function *f*
EFL_2
FL_3 and EFL_2 are integrated by function *f*
EFL_3
Recursive integration process ···
EFL_{n-1}
FL_n and $EFL_{(n-1)}$ are integrated by function *f*
EFL (= EFL_n)
J.247(08)_FA.10

Figure A.10 – Freeze occurrence example

Table A.1 – Coefficient table

i	p_i	q_i	r_i
2	0.03	1.99	1.33
3	0.06	1.53	2.11
4	0.13	1.06	2.83
5	0.16	9.01	1.34
6	0.18	15.38	–5.23
7	0.21	21.47	–11.33
8	0.24	24.84	–16.37

A.5 Subjective video quality estimation module

After the temporal/spatial feature parameters have been calculated in all the target processed video frames, the subjective video quality estimation module calculates the subjective video quality value. Below is the formulation of the objective measure of video quality, which is based on the five parameters defined above:

$$\begin{aligned} Q &= \alpha + \beta - f \times \alpha\beta + g, \\ \alpha &= aX_1 + bX_2 + cX_3 + dX_4, \text{ and} \\ \beta &= \mathrm{e} \times \log_{10}(X_5) \end{aligned} \tag{A-13}$$

where Q is the objective measure of video quality, X_1 is *PSNR*, X_2 is log*(–Min_HV)*, X_3 is *Ave_MEB*, X_4 is *FV_LME*, and X_5 is *EFL*. The coefficients *a*, *b*, *c*, *d*, *e*, *f* and *g* should be selected from Table A.2, depending on the video format.

Table A.2 – Coefficients for each resolution

	a	b	c	d	e	f	g
VGA	0.08902650	–0.50462008	–1.00336199	–0.01556439	–0.00130027	–280.38247290	2.13943898
CIF	0.10674316	–0.42102154	–0.95745108	–0.01931476	–0.00452231	–58.61923757	1.38258338
QCIF	0.11041146	–0.61015931	–1.37400776	–0.00123345	–0.12711221	–1.84263528	1.43376451

Annex B

Description of OPTICOM's video quality measure PEVQ

(This annex forms an integral part of this Recommendation)

B.1 Description of the OPTICOM model PEVQ

Perceptual evaluation of video quality (PEVQ) is a very robust model which is designed to predict the effects of transmission impairments on the video quality as perceived by a human subject. Its main targets are mobile applications and multimedia applications. PEVQ is built on PVQM, a TV quality measure developed by KPN and Swisscom. The key features of PEVQ are:

- (fast and reliable) temporal alignment of the input sequences based on multi-dimensional feature correlation analysis with limits that reach far beyond those tested by VQEG, especially with regard to the amount of time clipping, frame freezing and frame skipping which can be handled.
- Full frame spatial alignment.
- Colour alignment algorithm based on cumulative histograms.
- Detection and perceptually correct weighting of frame freezes and frame skips.
- Perceptual estimation of degradations.

Only five indicators are used to detect the video quality. Those indicators operate in different domains (temporal, spatial, luminance and chrominance) and are motivated by the human visual system (HVS). Perceptual masking properties of the HVS are modelled at several stages of the algorithm. These indicators are integrated using a sophisticated spatial and temporal integration algorithm.

In its first step, the algorithm performs all the alignment steps and information concerning frozen or skipped frames is collected. In the second step, the now synchronized and equalized images are compared for visual differences in the luminance as well as in the chrominance domain, taking masking effects and motion into account. This results in a set of indicators which all describe certain quality aspects. The last step is finally the integration of the indicators by non-linear functions in order to derive the final MOS.

Due to the low number of indicators and the resulting low degree of freedom, the model can hardly be over-trained and is very robust. PEVQ can be efficiently implemented without sacrificing accuracy.

B.1.1 Overview

The basic idea of the PEVQ model is given in Figure B.1. Since PEVQ is a full reference algorithm, it requires two input signals. One must be the undistorted source reference channel (SRC) S, and one must be the processed video sequence (PVS) P. It is assumed that the PVS is the result of the processed SRC through a hypothetical reference circuit (HRC). Both sequences must have the same resolution (e.g., VGA, CIF or QCIF). The signals should follow the YCbCr colour representation, as defined in [B7].

The data storage for the images is planar oriented, no down-sampling is used (sampling 4:4:4). Plane 1 is the Y component, plane 2 the Cb component and plane 3 the Cr component. The pixel values are stored as floating-point data.

In the pre-processing step, a spatial region of interest (ROI) is extracted from the reference and test signal. All subsequent calculations are performed on this ROI only, which is represented by the cropped signals S_P and P_P. The pre-processing is followed by a coarse alignment (registration) of the input sequences in the temporal and luminance domain. The "luminance and alignment information" obtained by these modules is used in the subsequent "detailed sequence alignment" process which performs the temporal frame-by-frame alignment of the two video sequences, a compensation for spatial shifts and a compensation for differences in colour and brightness based on histogram evaluations. The results of the "detailed sequence alignment" are the "matching info", which is used to determine the perceptual impact of temporal degradations, as well as the cropped and aligned sequences S_A and P_A.

The spatial distortions are further analysed by the "spatial distortion analysis" block, which calculates the perceptual differences between the sequences in the spatial domain, resulting in four distortion indicators.

The "matching info" is further processed by the perceptual "temporal distortion analysis", which results in one indicator representative for frame repeats and other temporal distortions.

In the last step of PEVQ, the five indicators that were derived above are weighted by logistic functions and combined to form the final PEVQ score, which correlates highly with a MOS obtained from subjective tests.

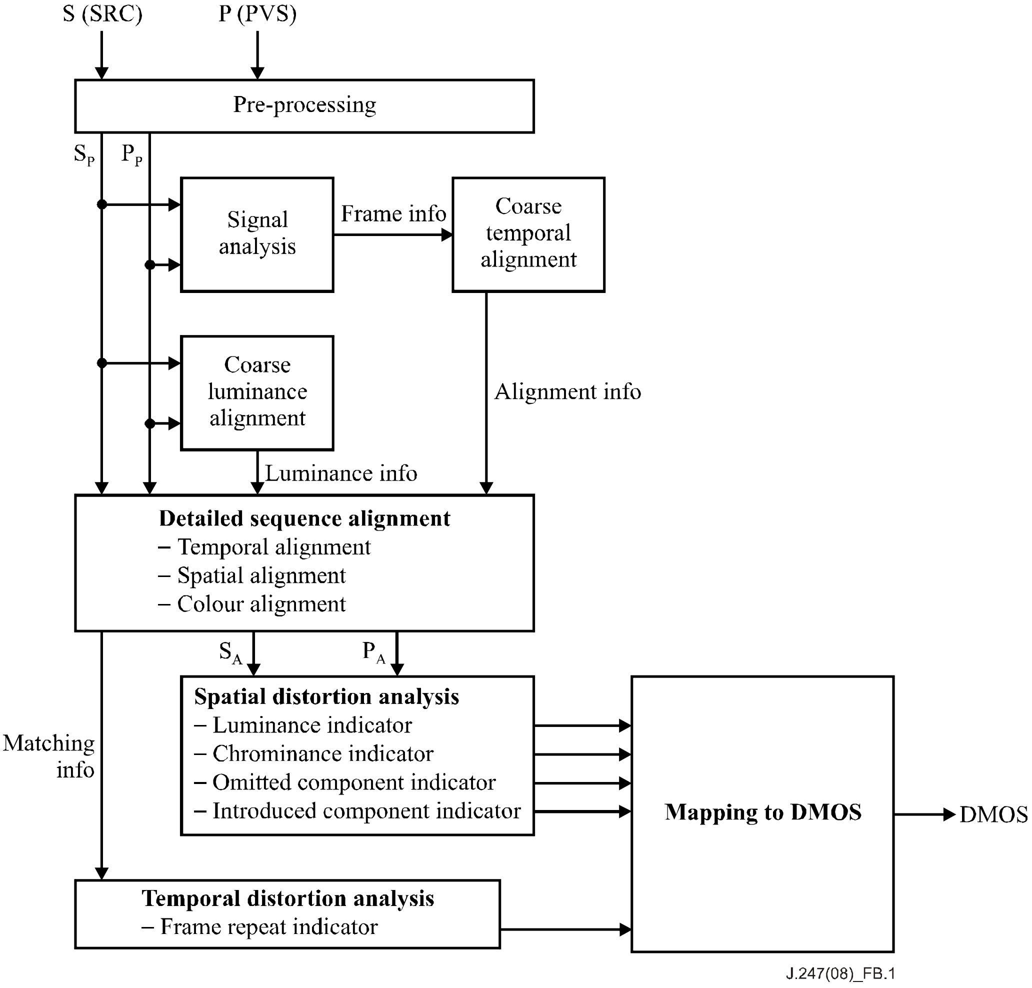


Figure B.1 – Overview of PEVQ

B.1.2 Pre-processing

Distortions nearest to the border are often ignored or not really noticed by viewers. Therefore, the input sequences are cropped to a region of interest calculated by:

$$S_P[i,j,t]=S[i+c,y+c,t] \quad \forall\, i\in[0..W-1-2c],\ j\in[0..H-1-2c] \qquad \text{(B-1)}$$

$$P_P[i,j,t]=P[i+c,y+c,t] \quad \forall\, i\in[0..W-1-2c],\ j\in[0..H-1-2c] \qquad \text{(B-2)}$$

Where W is the width of the input sequence S and H the height of the input sequence S.

The border value c depends on the size of the input sequence. For QCIF sequences c=3, for CIF sequences c=6, and for VGA sequences c=12.

B.1.3 Signal analysis

The task of the signal analysis part consists of the extraction of information and characterization of the processed signals. To avoid side effects, a border of 10 pixels around the image is removed:

$$C_s[i,j,t]=S_p[i+10,y+10,t] \quad \forall\, i\in\lfloor 0..W_{sp}-21\rfloor,\ j\in\lfloor 0..H_{sp}-21\rfloor \qquad \text{(B-3)}$$

$$C_P[i,j,t]=P_P[i+10,y+10,t] \quad \forall\, i\in\lfloor 0..W_{pp}-21\rfloor,\ j\in\lfloor 0..H_{pp}-21\rfloor \qquad \text{(B-4)}$$

In the next step, six different aspects in the spatial domain are extracted from the reference and the degraded signal, and stored in the matrix Ds, respectively Dp.

These are:

- Mean and standard deviations of the cropped signal:

$$D_S[0,t]=\frac{1}{W_{cp}H_{cs}}\sum_{i=0}^{W_{cs}-1}\sum_{j=0}^{H_{cs}-1}C_s[i,j,t] \qquad \text{(B-5)}$$

$$D_p[0,t]=\frac{1}{W_{cp}H_{cp}}\sum_{i=0}^{W_{cp}-1}\sum_{j=0}^{H_{cp}-1}C_p[i,j,t] \qquad \text{(B-6)}$$

$$D_s[1,t]=\sqrt{\frac{1}{W_{cs}H_{cs}}\sum_{i=0}^{W_{cs}-1}\sum_{j=0}^{H_{cs}-1}\left(C_s[i,j,t]-D_s[0,t]\right)^2} \qquad \text{(B-7)}$$

$$D_p[1,t]=\sqrt{\frac{1}{W_{cp}H_{cp}}\sum_{i=0}^{W_{cp}-1}\sum_{j=0}^{H_{cp}-1}\left(C_p[i,j,t]-D_P[0,t]\right)^2} \qquad \text{(B-8)}$$

- Mean and standard deviations of the difference between the cropped signal and the cropped signal shifted by 3 pixels in the horizontal and vertical direction:

$$D_S[2,t]=\frac{1}{(W_{cp}-3)(H_{cs}-3)}\sum_{i=0}^{W_{cs}-4}\sum_{j=0}^{H_{cs}-4}\left(C_s[i,j,t]-C_s[i+3,j+3,t]\right) \qquad \text{(B-9)}$$

$$D_P[2,t]=\frac{1}{(W_{cp}-3)(H_{cs}-3)}\sum_{i=0}^{W_{cs}-4}\sum_{j=0}^{H_{cs}-4}\left(C_P[i,j,t]-C_P[i+3,j+3,t]\right) \qquad \text{(B-10)}$$

$$D_s[3,t]=\sqrt{\frac{1}{(W_{cs}-3)(H_{cs}-3)}\sum_{i=0}^{W_{cs}-4}\sum_{j=0}^{H_{cs}-4}\left(C_s[i,j,t]-C_s[i+3,j+3,t]-D_s[2,t]\right)^2} \qquad \text{(B-11)}$$

$$D_p[3,t]=\sqrt{\frac{1}{(W_{cs}-3)(H_{cs}-3)}\sum_{i=0}^{W_{cp}-4}\sum_{j=0}^{H_{cp}-4}\left(C_p[i,j,t]-C_p[i+3,j+3,t]-D_P[2,t]\right)^2} \qquad \text{(B-12)}$$

For the aspects 5 and 6, edge images E_s[t] and E_p[t] are created by filtering the cropped images according to the following rule:

$$E_s[t]=\sqrt{(C_s[t]*K_h)^2+(C_s[t]*K_v)^2} \qquad \text{(B-13)}$$

$$E_p[t]=\sqrt{(C_p[t]*K_h)^2+(C_p[t]*K_v)^2} \qquad \text{(B-14)}$$

Where K_h and K_v are horizontal and vertical Sobel filter kernels:

$$K_h=\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix} \quad K_v=\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix} \qquad \text{(B-15)}$$

- The mean and standard deviation of the edge images $E_s[t]$ and $E_p[t]$ are chosen as informational aspects:

$$D_S[4,t]=\frac{1}{W_{cp}H_{cs}}\sum_{i=0}^{W_{cs}-1}\sum_{j=0}^{H_{cs}-1}E_s[i,j,t] \qquad \text{(B-16)}$$

$$D_p[4,t]=\frac{1}{W_{cp}H_{cp}}\sum_{i=0}^{W_{cp}-1}\sum_{j=0}^{H_{cp}-1}E_p[i,j,t] \qquad \text{(B-17)}$$

$$D_s[5,t]=\sqrt{\frac{1}{W_{cs}H_{cs}}\sum_{i=0}^{W_{cs}-1}\sum_{j=0}^{H_{cs}-1}\left(E_s[i,j,t]-D_s[4,t]\right)^2} \qquad \text{(B-18)}$$

$$D_p[5,t]=\sqrt{\frac{1}{W_{cp}H_{cp}}\sum_{i=0}^{W_{cp}-1}\sum_{j=0}^{H_{cp}-1}\left(E_p[i,j,t]-D_P[4,t]\right)^2} \qquad \text{(B-19)}$$

- In the time domain, the standard deviation of the difference between two consecutive frames is used in the later functions:

$$\overline{T_S}[t]=\frac{1}{W_{cs}H_{cs}}\sum_{i=0}^{W_{cs}-1}\sum_{j=0}^{H_{cs}-1}\left(C_s[i,j,t]-C_s[i,j,t-1]\right) \qquad \text{(B-20)}$$

$$\overline{T_P}[t]=\frac{1}{W_{cp}H_{cp}}\sum_{i=0}^{W_{cs}-1}\sum_{j=0}^{H_{cs}-1}\left(C_P[i,j,t]-C_P[i,j,t-1]\right) \qquad \text{(B-21)}$$

$$T_S[t]=\sqrt{\frac{1}{W_{cs}H_{cs}}\sum_{i=0}^{W_{cs}-1}\sum_{j=0}^{H_{cs}-1}\left(C_s[i,j,t]-C_s[i,j,t-1]-\overline{T_s}[t]\right)^2} \qquad \text{(B-22)}$$

$$T_p[t]=\sqrt{\frac{1}{W_{cp}H_{cp}}\sum_{i=0}^{W_{cp}-1}\sum_{j=0}^{H_{cp}-1}\left(C_p[i,j,t]-C_P[i,j,t-1]-\overline{T_P}[t]\right)^2} \qquad \text{(B-23)}$$

B.1.4 Fluidity estimate

B.1.4.1 Definition of the fluidity estimate vectors

The fluidity estimate is based on the temporal aspects and carried out for the reference and test signal separately. The results of the fluidity analysis will be required by the temporal alignment modules.

$$FE_s[t] = \begin{cases} 0 & if \quad T_s[t] > 0.1 \cdot mean(T_s[t]) \\ FE_s[t-1]+1 & else \end{cases} \qquad \text{(B-24)}$$

$$FE_s[0] = 0$$

$$FE_p[t] = \begin{cases} 0 & if \quad T_p[t] > 0.1 \cdot mean(T_p[t]) \\ FE_p[t-1]+1 & else \end{cases} \qquad \text{(B-25)}$$

$$FE_p[0] = 0$$

B.1.4.2 Maximum number of consecutive fluidity elements

The function *MaxFluidityElements*(*FE*,t_1,t_2) returns the maximum number of consecutive non-zero elements of *FE*[*t*] which lie between t_1 and t_2.

B.1.5 Coarse temporal alignment

An important part of a full reference algorithm is the temporal registration of the received video sequence to the given reference sequence. The temporal alignment may be corrupted by delay, frame skipping or repetition. In PEVQ, this alignment procedure is done in two steps. In the first step, a coarse alignment together with an estimate of the accuracy of the coarse alignment results is done, followed by a fine alignment module.

PEVQ uses a top down strategy which is based on finding an optimum cross-correlation between the spatial information aspects. The algorithm starts by searching a static delay for the entire sequences. If no delay can be found which fulfils some reliability criteria, the signals are split into two parts and the procedure is repeated for each segment separately. This is continued until all segments could be aligned accurately enough or a minimum segment length is reached. To perform the actual alignment, six different parameters are calculated for each frame of the reference and the degraded signal. The resulting vectors are analysed and the correlation between them is calculated. The best suitable result is then selected to determine the coarse delay for the segment.

B.1.5.1 Similarity measure

Let $X[t]$ and $Y[t]$ be two vectors with the length *nx* and *ny*.

Then, a normalized product-moment correlation is given by:

$$\tilde{r}_{xy}[n] = \frac{1}{\sqrt{\sum_{t=0}^{nx-1}\left(X[t]-\bar{\chi}\right)\sum_{t=0}^{ny-1}\left(Y[t]-\bar{\xi}\right)}} \sum_{t=-nx}^{nx}\left(X[t]-\bar{\chi}\right)\left(Y[t+n]-\bar{\xi}\right) \qquad \text{(B-26)}$$

with

$$\bar{\chi} = \frac{1}{xStop - xStart + 1}\sum_{t=xStart}^{xStop} X[t] \qquad \text{(B-27)}$$

where *xStart* is the index of the first and *xStop* the index of the last non-zero value within the $X[t]$ vector.

and

$$\bar{\xi} = \frac{1}{yStop - yStart + 1} \sum_{t=yStart}^{yStop} Y[t] \qquad \text{(B-28)}$$

where *yStart* is the index of the first and *yStop* the index of the last non-zero value within the $Y[t]$ vector.

An estimate of the delay between the vectors is given by:

$$delay(X[t], Y[t]) = \underset{n}{\arg\max}(\tilde{r}_{xy}[n]) \qquad \text{(B-29)}$$

The value

$$corrm(X[t], Y[t]) = \max_{n}(\tilde{r}_{xy}[n]) \qquad \text{(B-30)}$$

is the similarity measure and represents an estimate of the accuracy of the estimated delay.

B.1.5.2 Definitions of the similarity measure for the spatial information aspects

To calculate the delay estimation, PEVQ uses the spatial information aspects, stored in the matrices D_s and D_p, as defined in clause B.1.3. These types of matrices are referred to as similarity matrices.

B.1.5.2.1 Similarity measure

The similarity measure of two similarity matrices is defined as the maximum similarity measure of the individual information aspects.

$$corrmsim(D_s, D_p) = \max_{i \in [0,5]}\left(corrm\left(D_s[i,t], D_p[i,t]\right)\right) \qquad \text{(B-31)}$$

B.1.5.2.2 Similarity index

The similarity index of two similarity matrices is defined as the index of the information aspect with the highest similarity measure.

$$corrmx(D_s, D_p) = \underset{i \in [0,5]}{\arg\max}\left(corrm\left(D_s[i,t], D_p[i,t]\right)\right) \qquad \text{(B-32)}$$

B.1.5.2.3 Delay between similarity matrices

The delay between two similarity matrices is defined as the delay of the information aspect with the highest similarity measure.

$$delaysim(D_s, D_p) = delay\left(D_s\left[corrmx\left(D_s, D_p\right)\right], D_p\left[corrmx\left(D_s, D_p\right)\right]\right) \qquad \text{(B-33)}$$

B.1.5.2.4 Realign procedure

The realign procedure of a similarity matrix *D*[*j,t*] is defined as follows:

$$realign(D_s, NumFrames) = \begin{cases} \begin{array}{l} \textit{shift each element of } D_s \\ \textit{NumFrames columns to the} \\ \textit{right, discarding the last} \\ \textit{NumFrames columns.} \\ \textit{Set the elements in the first} \\ \textit{NumFrames columns of } D_s \\ \textit{to zero.} \end{array} & \textit{If NumFrames} >= 0 \\ \\ \begin{array}{l} \textit{shift each element of } D_s \\ \textit{NumFrames columns to the} \\ \textit{left, discarding the first} \\ \textit{NumFrames columns.} \\ \textit{Set the elements in the last} \\ \textit{NumFrames columns of } D_s \\ \textit{to zero.} \end{array} & \textit{If NumFrames} < 0 \end{cases} \qquad \text{(B-34)}$$

B.1.5.2.5 Extract region of interest

The function $extr(D, first, last)$ creates a new similarity matrix *E* out of the similarity matrix *D* following the rule:

$$E[i,t] = D[i, t + first], \ \forall \ i \in \{0..5\}, t \in \{0, last - first - 1\} \qquad \text{(B-35)}$$

B.1.5.2.6 Length of the similarity matrix

The function $length(D)$ returns the number of rows of the similarity matrix *D*.

B.1.5.3 Actual coarse temporal alignment

The inputs to the coarse alignment process are the spatial similarity matrices of the degraded and the reference sequences D_S and D_P with 6x*L* elements, with *L* being:

$$L = \max(\textit{Nr of frames in PVS}, \textit{Nr of frames in SRC}). \qquad \text{(B-36)}$$

Any missing elements in the shorter vector are zero padded. The outcome of this process is a frame-wise estimate of the delay and an estimate of the accuracy of the delay.

The coarse alignment creates a tree of submatrices by iteratively splitting each similarity and submatrix in two halves, until each submatrix contains between 8 and 16 columns.

For each submatrix the delay information Del, the frame number S where the segment starts and the length *L* of the segments are stored in a *DelayInfo* structure.

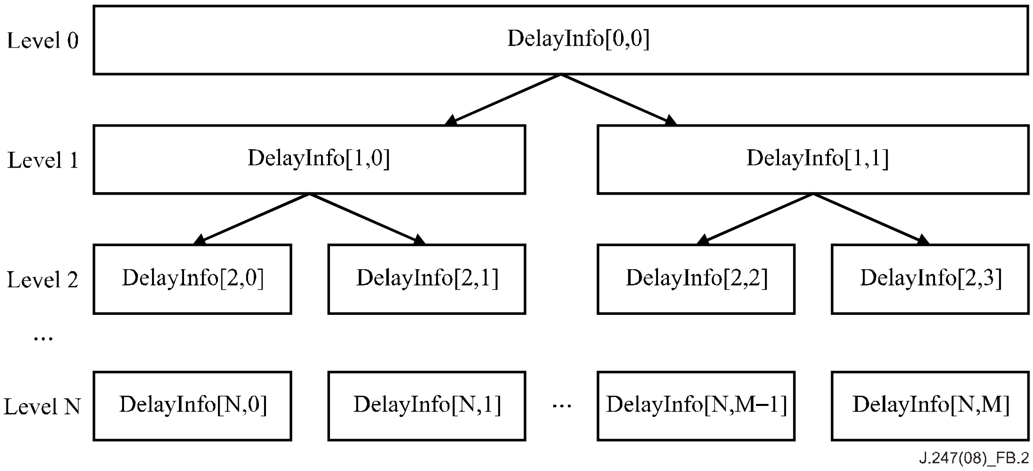


Figure B.2 – Submatrix Tree

This procedure is performed using the following steps:

Step 1: Raw global alignment:

- Estimate an overall delay between the zeros padded similarity matrices:

$$\mathrm{Del} = \mathit{delaysim}(Dp,Ds)$$

- Check whether the delay is within the maximum allowed range, otherwise force it to 0:

$$\text{If } Abs(Del>(0.25 * Fps+1))\ Del = 0$$

- Assign $DelayInfo[0,0].Del = Del$ as delay for the matrix
- Assign $DelayInfo[0,0].L = length(Dp)$ for the matrix
- Assign $DelayInfo[0,0].S = 0$ for the first column in the matrix

Step 2: Refine delay estimate using a tree approach:

Apply the following to all submatrices from level 0 to level N in Figure B.2 and from left to right:

- Let i be the level of the input segment
- Let j be the index of the input segment
- Let $L = DelayInfo[i,j].L$ be the length of the input matrices
- Let $Del = DelayInfo[i,j].Del$ the delay between the input matrices
- Let $S = DelayInfo[i,j].S$ be the frame number assigned to the first column in the input matrix
- Re-align the similarity matrix Ds:

$$Ds'=realign(Ds,Del)$$

- Create two new submatrices (left matrix and right matrix) by splitting the input matrices in two parts with an overlap of 50% at each side;

  which results in:

$$Dp'[Left] = extr(Dp,–L/4+S,L/2+L/4+S)$$

$$Ds'[Left] = extr(Ds',–L/4+S,L/2+L/4+S)$$

  for the left side, and:

$$Dp'[Right] = extr(Dp,–L/4+L/2+S,L+L/2+S)$$

$$Ds'[Right] = extr(Ds',–L/4+L/2+S,L+L/2+S)$$

for the right side:

- Calculate the delays between each pair of matrices:

$$Del'[Left] = delaysim(Dp'[Left], Ds'[Left]) + Del'$$

$$Del'[Right] = delaysim(\mathrm{Dp'[Right]}, Ds'[Right]) + Del'$$

For the left side matrix assign:

- $DelayInfo[i+1,2j].Del = Del'[Left]$ as delay for the matrix
- Assign $DelayInfo[i+1,2j].L = floor(L/2)$ as length for the matrix
- Assign $DelayInfo[i+1,2j].S = S$ as the frame number for the first column in the matrix

For the right side matrix assign:

- $DelayInfo[i+1,2j+1].Del = Del'[Right]$ as delay for the matrix
- Assign $DelayInfo[i+1,2j+1].L = ceil(L/2)$ as length for the matrix
- Assign $DelayInfo[i+1,2j+1].S = S+ceil(L/2)$ as the frame number for the first column in the matrix

Step 3:

- Repeat step 2 until the length of all resulting submatrices is less than 16.

B.1.5.3.1 Specification for raw matching vector

A vector *RawMatch[t]* which, for each frame in the PVS sequence, assigns a corresponding frame from the SRC sequence, is calculated as described in the pseudo code below:

```
For(j=0; j<M-1; j++)
{
    For (k= DelayInfo[N,j].S; k<DelayInfo[N,j+1].S; k++)
    {
        RawMatch[k]=k+DelayInfo[N,j].Del;
    }
}
For (k= DelayInfo[N,M].S;k< DelayInfo[N,M].S +DelayInfo[N,M].L; k++)
{
        RawMatch[k ]= k +DelayInfo[N,M].Del;
}
```

B.1.5.3.2 Specification of first and last index

The index of the first and the last frame in the PVS which can be assigned to the corresponding frame from SRC is calculated as follows:

```
StartMatchIndex=0
While(RawMatch[StartMatchIndex]<0)StartMatchIndex++

StopMatchIndex=Nr frames Degraded-1
While(RawMatch[StopMatchIndex]>=Nr frames Degraded)StopMatchIndex--
```

B.1.5.3.3 Specification of the fine alignment search range vector

For each index *j* of Level *N*, the search range value is assigned by the following steps:

```
Search[j] = 0
Search[j] = max(Search[j], abs(DelayInfo[N,j].Del- DelayInfo[N,j-1].Del))
Search[j]= max(Search[j], abs(DelayInfo[N,j].Del- DelayInfo[N,j+1].Del))
Search[j] = max(Search[j], abs((DelayInfo[N,j].S+ DelayInfo[N,j].L)-
DelayInfo[N,j+1].S))
Search[j] = max(Search[j], abs((DelayInfo[N,j-1].S+ DelayInfo[N,j-1].L)-
DelayInfo[N,j].S))
Search[j] = max(Search[j], MaxFluidityElements(FR_S,DelayInfo[N,j].S, DelayInfo[N,j].L))
Search[j] = max(Search[j], MaxFluidityElements(FR_P,DelayInfo[N,j].S, DelayInfo[N,j].L))
```

A vector *SearchRange*[*t*] for each frame may be calculated as described by the pseudo code below:

```
For(j=0; j<M-1; j++)
{
        For (k= DelayInfo[N,j].S; k<DelayInfo[N,j+1].S  ; k++)
        {
            SearchRange[k ]= Search[j];
        }
}
For (k= DelayInfo[N,M].S; k< DelayInfo[N,M].S+ DelayInfo[N,M].L; k++)
{
        SearchRange[k ]= Search[j];
}
```

B.1.6 Coarse luminance alignment

B.1.6.1 Determination of the coarse luminance alignment correction curve

A preliminary luminance alignment, based on the luminance part of the temporally unaligned SRC and PVS, is carried out.

First, the histograms of the luminance components are computed by:

$$h_{s,y}[k]=\frac{1}{N\cdot W\cdot H}\sum_{t=0}^{N-1}\sum_{i=0}^{H-1}\sum_{j=0}^{W-1}\delta\left[k,S_{p,y}[i,j,t]\right] \qquad \text{(B-37)}$$

$$h_{p,y}[k]=\frac{1}{N\cdot W\cdot H}\sum_{t=0}^{N-1}\sum_{i=0}^{H-1}\sum_{j=0}^{W-1}\delta\left[k,P_{p,y}[i,j,t]\right] \qquad \text{(B-38)}$$

with

$$\delta[a,b]=\begin{cases}1 & if \quad a=b\\ 0 & otherwise\end{cases} \qquad \text{(B-39)}$$

In the next step, the cumulative histograms are calculated:

$$HC_{s,y}[\lambda] = \sum_{k=0}^{\lambda} h_{s,y}[k] \tag{B-40}$$

$$HC_{P,y}[\lambda] = \sum_{k=0}^{\lambda} h_{P,y}[k] \tag{B-41}$$

The luminance correction *CorrectionCurve*$_y$, which aligns the luminance part of the temporally unaligned SRC and PVS, is calculated as described in the following pseudo code:

```
Int   binS      = 0
Int   binP      = 128

Float fracS     = $h_{s,y}$[binS]
Float cumFracS  = $HC_{s,y}$[binS]
Float fracP     = $h_{p,y}$[binP]
Float cumFracP  = $HC_{p,y}$[binP]

Int tarS        = 0
Int SteepnessS  = 0

while(($HC_{s,y}$[binS]<= cumFracP) && (binS<255)
{
  binS++
}



for (binP = 128; binP < 256; binP++)
{
     fracP=$h_{p,y}$[binP]
     cumFracP = $HC_{p,y}$[binP]
     if (binS < 255)
     {
          if ((fracS < 0.0008) && (fracP < 0.0008))
          {
               binS++
               fracS = $h_{s,y}$[binS]
               cumFracS = $HC_{s,y}$[binS]

               tarS=binS

          }
          else
          {
               SteepnessS=0
               while ((cumFracS < cumFracP)&&
                         (binS < 255) && (SteepnessS <= 50)
               {
                    binS++
                    SteepnessS++
                    fracS = $h_{s,y}$[binS]
                    cumFracS = $HC_{s,y}$[binS]
               }
               if(cumFracS >= cumFracP)
               {
                tarS=(binS-1)*( $HC_{s,y}$[binS]- cumFracP)+
                            (binS)*((cumFracP- $HC_{sy}$[binS-1])/
                                     ( $HC_{s,y}$[binS]- $HC_{s,y}$[binS-1]))

                tarS=max(tarS,binS-1)
                tarS=min(tarS,binS)

            }
            else
            {
              tarS=binS
            }

        }
    }
    $CorrectionCurve_{y}$[binP] = round(tarS)
}

binS = 255
fracS = $h_{s,y}$[binS]
cumFracS = $HC_{s,y}$[binS]


cumFracP = $HC_{p,y}$[128]
while(($HC_{s,y}$[binS]>= cumFracP) && (bins>0)
{
  binS--
}

for (binP = 127; binP >= 0; binP--)
{
    fracP=$h_{p,y}$[binP]
    cumFracP = $HC_{p,y}$[binP]
    if (binS >= 0)
    {
        if ((fracS < 0.0008) && (fracP < 0.0008))
        {
            binS--
            fracS = $h_{s,y}$[binS]
            cumFracS = $HC_{s,y}$[binS]
            tarS=binS
        }
        else
        {
            Steepness=0
            while ((cumFracS > cumFracP) &&
                    (binS >= 0)&&
                    (Steepness<=50))
            {
                bins--
                Steepness++
                fracS = $h_{s,y}$[binS]
                cumFracS = $HC_{s,y}$[binS]
            }

            if(cumFracS <= cumFracP)
            {
                tarS=(binS)*( $HC_{s,y}$[bins+1]- cumFracP)+
                          (binS+1)*((cumFracP- $HC_{s,y}$[binS])/
                                 ( $HC_{s,y}$[binS+1]- $HC_{s,y}$[binS]))

                tarS=max(tarS,binS)
                tarS=min(tarS,bins+1)


            }
            else
            {
              tarS=binS
            }
        }
    }
    $CorrectionCurve_y$[binP] = round(tarS)
}
```

B.1.6.2 Histogram correction

The histogram correction is carried out by applying the *CorrectionCurve*$_y$ as a table lookup for each PVS component.

$$S_{c,y}[i,j,t]=S_{p,y}[i,j,t] \tag{B-42}$$

$$P_{c,y}[i,j,t]=CorrectionCurve_y\left[P_{p,y}[i,j,t]\right] \tag{B-43}$$

B.1.7 Fine temporal alignment

B.1.7.1 Determination of the temporal registration vector

The temporal registration of the received video sequence is done using an MSE criterion, and carried out on the coarse luminance-aligned sequences.

By minimizing the function:

$$f\left(\delta_x,\delta_y,t,RawMatch[t]+\delta_t[t]\right)\text{–>}\min \tag{B-44}$$

where

$$f\left(\delta_x,\delta_y,t_p,t_s\right)=\sqrt{\frac{1}{Norm}\sum_{i=\max(0,\delta_x)}^{\min(W,W+\delta_x)-1}\;\sum_{j=\max(0,\delta_y)}^{\min(H,H+\delta_y)-1}\left|P_{c,y}\left(i+\delta_x,j+\delta_y,t_p\right)-S_{c,y}\left(i,j,t_s\right)\right|^2} \tag{B-45}$$

with

$$Norm=\left(\min(W,W+\delta_x)-\max(0,\delta_x)\right)\left(\min(W,W+\delta_y)-\max(0,\delta_y)\right)$$

Over the three variables δ_x,δ_y and δ_t:

$$\delta_x\in\{-1,0-1\},\delta_y\in\{-1,0,1\},\delta_t[t]\in\{-SearchRange[t],\ldots,SearchRange[t]\} \tag{B-46}$$

Over the frames which satisfy:

$$StartMatchIndex\le t<StopMatchIndex \tag{B-47}$$

The values δ_{x0}, δ_{y0} and δ_{t0} are determined. Note that, in case that the *SearchRange*[t] is 0, δ_{t0} is known to be 0 in advance.

A matching vector which contains the best match between the reference and test images is given by:

$$MatchingPictureIndex[t] = RawMatch[t] + \delta_{t0}[t] - FE_s[RawMatch[t] + \delta_{t0} + [t]] \qquad \text{(B-48)}$$

The matching picture index is smoothed according to the following recursion:

$$SmoothedMatching[t] = \begin{cases} MatchingPictureIndex[t] & if \quad FE_p[t] = 0 \\ SmoothedMatching[t-1]+1 & else \end{cases} \qquad \text{(B-49)}$$

B.1.7.2 Temporal alignment correction

The temporal correction is carried out on the reprocessed sequences:

$$S_{t,\mu}[i,j,t] = S_{p,\mu}[i,j,SmoothedMatching[t + StartMatchIndex]] \qquad \text{(B-50)}$$

$$P_{t,\mu}[i,j,t] = P_{p,\mu}[i,j,t + StartMatchIndex]$$

Where $\mu \in \{Y, Cr, Cb\}$ represents the different colour planes.

B.1.8 Spatial alignment

B.1.8.1 Determination of the spatial offset

The same mean squared error approach, as in the temporal registration, is used for determination of the spatial shift between the reference and test signals on a frame-by-frame basis. The alignment is carried out on the temporally aligned signals.

The spatial offsets are determined by minimizing the function:

$$f(\delta_x[t], \delta_y[t], t) -> \min \qquad \text{(B-51)}$$

where

$$f(\delta_x, \delta_y, t) = \sqrt{\frac{1}{Norm} \sum_{i=\max(0,\delta_x)}^{\min(W,W+\delta_x)-1} \sum_{j=\max(0,\delta_y)}^{\min(H,H+\delta_y)-1} \left| P_t(i+\delta_x, j+\delta_y, t) - S_t(i,j,t) \right|^2} \qquad \text{(B-52)}$$

and

$$Norm = (\min(W, W+\delta_x) - \max(0,\delta_x))(\min(W, W+\delta_y) - \max(0,\delta_y))$$

over the two variables δ_x, δ_y. The minimum values $\delta_{x0}[t], \delta_{y0}[t]$ represent the spatial offset between the reference and test sequence.

B.1.8.2 Spatial offset correction

The spatial corrected sequences are calculated by:

$$S_{s,\mu}[i,j,t] = S_{t,\mu}[i,j,t] \qquad \text{(B-53)}$$

$$P_{s,\mu}[i,j,t] = \begin{cases} P_{t,\mu}[i+\delta_{x0}[t], j+\delta_{y0}[t], t] & if \ 0 \le (i+\delta_{x0}[t]) < W \ and \ 0 \le (j+\delta_{y0}[t]) < H \\ S_{s,\mu}[i,j,t] & else \end{cases} \qquad \text{(B-54)}$$

where $\mu \in \{Y, Cr, Cb\}$ represents the different colour planes.

B.1.9 Fine luminance and chrominance alignment

B.1.9.1 Determination of the correction curves

The first step is to compute the histograms of the reference and distorted image for the luminance and the chrominance components. Let S_s represent the temporal and spatial aligned reference image, and P_s the temporal and spatial aligned distorted image, and then the histogram is calculated as follows:

$$h_{s,\mu}[k]=\frac{1}{N\cdot W\cdot H}\sum_{t=0}^{N-1}\sum_{i=0}^{H-1}\sum_{j=0}^{W-1}\delta[k,S_s[i,j,t]] \tag{B-55}$$

$$h_{P,\mu}[k]=\frac{1}{N\cdot W\cdot H}\sum_{t=0}^{N-1}\sum_{i=0}^{H-1}\sum_{j=0}^{W-1}\delta[k,P_\mu[i,j,t]] \tag{B-56}$$

with

$$\delta[a,b]=\begin{cases}1 & if \quad a=b\\ 0 & otherwise\end{cases} \tag{B-57}$$

where $\mu\in\{Y,Cr,Cb\}$ represents the different colour planes.

In the next step, the cumulative histograms of the reference and distorted signals are calculated:

$$HC_{s,\mu}[\lambda]=\sum_{k=0}^{\lambda}h_{s,\mu}[k] \tag{B-58}$$

$$HC_{P,\mu}[\lambda]=\sum_{k=0}^{\lambda}h_{P,\mu}[k] \tag{B-59}$$

The aim of the histogram correction is to align the histograms of the processed video sequence as closely as possible to those of the source sequence. Assuming that the cumulative histograms of the source sequence and the distorted sequence are strictly increasing so that their inverse exists, the transformation function may be found as:

$$Z_\mu[k]=HC_{S,\mu}^{-1}\left(HC_{P,\mu}^{-1}[k]\right) \tag{B-60}$$

In practice, the values at the upper and lower range are clipped to a certain threshold. In addition, e.g., due to coding artefacts, some histogram values may be zero, causing the cumulative histogram to be locally non-invertible. Therefore, the transformation formula *Z* may not be directly applied.

The approach used in PEVQ is a combination of using the transformation formula for *Z* whenever there is enough empirical probability mass and using a straight line computation elsewhere.

The following pseudo C code illustrates the computation of the luminance correction curve *CorrectionCurve*$_y$:

```
Int   binS      = 0
Int   binP      = 128

Float fracS     = $h_{s,y}$[binS]
Float cumFracS  = $HC_{s,y}$[binS]
Float fracP     = $h_{p,y}$[binP]
Float cumFracP  = $HC_{p,y}$[binP]

Int tarS        = 0
Int SteepnessS = 0

while(($HC_{sy}$[binS]<= cumFracP) && (binS<255)
{
    binS++
}

for (binP = 128; binP < 256; binP++)
{
    fracP=$h_{p,y}$[binP]
    cumFracP = $HC_{p,y}$[binP]
    if (binS < 255)
    {
        if ((fracS < 0.0008) && (fracP < 0.0008))
        {
            binS++
            fracS = $h_{s,y}$[binS]
            cumFracS = $HC_{s,y}$[binS]

            tarS=binS

        }
        else
        {
            SteepnessS=0
            while ((cumFracS < cumFracP)&&
                   (binS < 255) && SteepnessS <= 50)
            {
                binS++
                SteepnessS++
                fracS = $h_{s,y}$[binS]
                cumFracS = $HC_{s,y}$[binS]
            }
            if(cumFracS >= cumFracP)
            {

                tarS=(binS-1)*( $HC_{s,y}$[binS]- cumFracP)+
                           (binS)*((cumFracP- $HC_{sy}$[binS-1])/
                                   ( $HC_{s,y}$[binS]- $HC_{s,y}$[binS-1]))

                tarS=max(tarS,binS-1)
                tarS=min(tarS,binS)

            }
            else
            {
              tarS=binS
            }

        }
    }
    CorrectionCurve$_{y}$[binP] = round(tarS)
}

binS = 255
fracS = $h_{s,y}$[binS]
cumFracS = $HC_{s,y}$[binS]
```

```
cumFracP = $HC_{p,y}$[128]
while(($HC_{s,y}$[binS]>= cumFracP) && (bins>0)
{
    binS--
}

for (binP = 127; binP >= 0; binP--)
{
    fracP=$h_{p,y}$[binP]
    cumFracP = $HC_{p,y}$[binP]
    if (binS >= 0)
    {
        if ((fracS < 0.0008) && (fracP < 0.0008))
        {
            binS--
            fracS = $h_{s,y}$[binS]
            cumFracS = $HC_{s,y}$[binS]
            tarS=binS
        }
        else
        {
            Steepness=0
            while ((cumFracS > cumFracP) &&
                        (binS >= 0)&&
                        (Steepness<=50))
            {
                bins--
                Steepness++
                fracS = $h_{s,y}$[binS]
                cumFracS = $HC_{s,y}$[binS]
            }

            if(cumFracS <= cumFracP)
            {

                tarS=(binS)*( $HC_{s,y}$[bins+1]- cumFracP)+
                            (binS+1)*((cumFracP- $HC_{s,y}$[binS])/
                                    ( $HC_{s,y}$[binS+1]- $HC_{s,y}$[binS]))

                tarS=max(tarS,binS)
                tarS=min(tarS,bins+1)


            }
            else
            {
              tarS=binS
            }
        }
    }
    CorrectionCurve$_{y}$[binP] = round(tarS)
}
```

Due to the fact that the source sequence may be colourless, whereas the processed sequence contains cross colour artefacts, a minimum slope of the correction curve is required. Therefore the calculation of the chrominance correction curves is slightly different to that of the luminance curve.

This is presented in the following pseudo C code, where:

$$h_{sc} \in \{h_{s,Cr}, h_{s,Cb}\} HC_{sc} \in \{HC_{s,Cr}, HC_{s,Cb}\} \quad h_{pc} \in \{h_{P,Cr}, h_{P,Cb}\} HC_{pc} \in \{HC_{P,Cr}, HC_{P,Cb}\}$$

and

$$CorrectionCurve_c \in \{CorrectionCurve_{Cb}, CorrectionCurve_{Cr}\}$$

should be set according to the processed chrominance plane.

```
Int   binS      = 0
Int   binP      = 128
Float fracS     = $h_{s,c}$[binS]
Float cumFracS = $HC_{s,c}$[binS]
Float fracP     = $h_{s,c}$[binP]
Float cumFracP = $HC_{s,c}$[binP]
Float tarS      = 0
Int   oldBinS = binS
Float steps     = 0

        while(($HC_{s,c}$[binS]<= cumFracP) && (binS<255)
        {
        binS++
}

for (binP = 128; binP < 256; binP++)
{
        Steps = Steps+0.5

        fracP=$h_{p,c}$[binP]
        cumFracP = $HC_{p,c}$[binP]

        if (binS < 255)
        {
            if ((fracS < 0.0008) && (fracP < 0.0008))
            {
                binS++
                fracS = $h_{sc}$[binS]
                cumFracS = $HC_{s,c}$[binS]
                tarS=binS
            }
            else
            {
                while ((cumFracS < cumFracP)&& (binS < 255))
                 {
                     binS++
                     fracS = $h_{sc}$[binS]
                     cumFracS = $HC_{s,c}$[binS]
                }
                if(cumFracS >= cumFracP)
                {

                     tarS=(binS-1)*( $HC_{s,c}$[binS]- cumFracP)+
                                             (binS)*((cumFracP- $HC_{s,c}$[binS-1])/
                                                  ( $HC_{s,c}$[binS]- $HC_{s,c}$[binS-1]))
                     tarS=max(tarS,binS-1)
                     tarS=min(tarS,binS)

                }
                else
                {
                tarS=binS
                }
           }
      }

      if (steps >= 1)
      {
           if ((binS – oldBinS)/steps < 1)
           {
                binS++
                tars++
           }
           steps = 0
           oldBinS = binS
      }
      CorrectionCurve_c[binP] = round(tarS)
}
binS = 255
fracS = $h_{sc}$[binS]
cumFracS = $HC_{s,c}$[binS]
oldBinS = binS
```

```
steps=0

cumFracP = $HC_{p,c}$[128]
    while(($HC_{s,c}$[binS]>= cumFracP) && (binS>0) binS--

for (binP = 127; binP >= 0; binP--)
{
        Steps=Steps+0.5
        fracP=$h_{pc}$[binP]
        cumFracP = $HC_{p,c}$[binP]
        if (binS >= 0)
        {
            if ((fracS < 0.0008) && (fracP < 0.0008))
            {
                binS--
                fracS = $h_{sc}$[c,binS]
                cumFracS = $HC_{s,c}$[c,binS]
            }
            else
            {
                while ((cumFracS > cumFracP) && (binS >= 0))
                {
                    binS--
                    fracS = $h_{sc}$[binS]
                    cumFracS = $HC_{s,c}$[binS]
                }

                if(cumFracS <= cumFracP)
                {

                    tarS=(binS)*( $HC_{s,c}$[bins+1]- cumFracP)+
                                    (binS+1)*((cumFracP- $HC_{s,c}$[binS])/
                                            ( $HC_{s,c}$[binS+1]- $HC_{s,c}$[binS]))

                    tarS=max(tarS,binS)
                    tarS=min(tarS,binS+1)

                }
                else
                {
                  tarS=binS
                }


            }
        }
        if (steps >= 1)
        {
            if ((oldBinS-binS)/steps < 1)
            {
                binS++
                tarS++
            }
            steps = 0
            oldBinS = binS
        }
        CorrectionCurve$_c$[binP] =round(tarS)
}
```

B.1.9.2 Histogram correction

To carry out the histogram correction on the raw processed sequence is computationally simple. It involves a simple table lookup for each component:

$$S_{A,\mu}[i,j,t] = S_{s,\mu}[i,y,t] \qquad \text{(B-61)}$$

$$P_{A,\mu}[i,j,t] = CorrectionCurve_{\mu}[P_S[i,y,t]] \qquad \text{(B-62)}$$

where $\mu \in \{Y, Cr, Cb\}$ represents the different colour planes

B.1.10 Spatial distortion analysis

In the spatial domain, four different indicators, one describing the difference in edginess of the luminance signal, one the difference in edginess in the chrominance domain, and two that describe the amount of movement in the video sequence, are calculated. The following clauses describe these indicators in detail.

B.1.10.1 Edginess image

The edginess of the reference and degraded image is computed as an approximation of the local gradient of the luminance and chrominance components of the aligned signals.

$$S_{edge,\mu}[t] = MaxFilter_{3x3}(\sqrt{\left(S_{A,\mu}[t] * K_v\right)^2 + \left(S_{A,\mu}[t] * K_h\right)^2} \qquad \text{(B-63)}$$

$$P_{edge,\mu}[t] = MaxFilter_{3x3}(\sqrt{\left(P_{A,\mu}[t] * K_v\right)^2 + \left(P_{A,\mu}[t] * K_h\right)^2} \qquad \text{(B-64)}$$

with the vertical filter kernel:

$$K_v = [0.5,0.5,0,-0.5,-0.5]$$

and the horizontal filter kernel:

$$K_h = [0.5,0.5,0,-0.5,-0.5]^T$$

where $\mu \in \{Y, Cr, Cb\}$ represents the different colour planes.

MaxFilter$_{3x3}$ () performs a non-linear filtering operation on the image. The function sets each pixel in the destination image to the maximum value of the nine source image pixel values in the neighbourhood of the processed pixel.

B.1.10.2 Luminance indicator

The luminance indicator is calculated based on the luminance part of the edge images.

Subjective experiments are normally carried out with a background illumination that corresponds to the mid-range grey level of the display. The HVS of the test persons adapts to this background illumination. As a result, the visual qualities of dark and light areas in the picture are of lesser importance to the quality judgement. In order to reflect this effect, the deviation of the luminance from 100 is computed for both the source video signal and the processed video signal. It is the maximum of these deviations that is used in the edginess frame indicator.

$$dev[i,j,t] = \max\left(\left|S_{A,y}[i,j,t] - 100\right|, \left|P_{A,y}[i,j,t] - 100\right|\right) \tag{B-65}$$

If, for a particular pair of source and processed sequences, wherever S_{edge} is non-zero, P_{edge} is smaller than S_{edge}', this is perceived as a loss of sharpness. Some source sequences have an overall higher edginess than others. It turns out that the *relative* decrease of P_{edge}' with respect to S_{edge}' is a better indicator of the loss of sharpness than the absolute difference. The indicator computed in this clause not only indicates loss of sharpness (P_{edge} $[i,j,t]$ minus $S_{edge}[i,j,t]$ negative). Also, the introduction of sharpness is registered as a distortion ($P_{edge}[i,j,t]$ minus $S_{edge}[i,j,t]$ positive). The relativeness of the effect also manifests itself locally for introduced edginess. The introduction of edginess in areas with a lot of edginess is less disturbing than the introduction of sharpness where little edginess is originally present.

$$e_Y[i,j,t] = \frac{P_{edge,Y}[i,j,t] - S_{edge,Y}[i,j,t]}{S_{edge,Y}[i,j,t] + 80 + dev[i,j,t]} \tag{B-66}$$

The normalized change in edginess *e* is locally clipped to the range [–40,40].

$$e_{clipped,Y}[i,j,t] = \begin{cases} -40 & if \quad e_Y[i,j,t] \le -40 \\ 40 & if \quad e_Y[i,j,t] \ge 40 \\ e_Y[i,j,t] & otherwise \end{cases} \tag{B-67}$$

The aggregation over space is done using a weighted L5 norm of the clipped change in edginess image:

$$e_Y[t] = \sqrt[5]{\frac{\sum_{i=0}^{W-1}\sum_{j=0}^{H-1} \left|e_{clipped,Y}[i,j,t]\right|^5 w[i,j]}{\sum_{i=0}^{W-1}\sum_{j=0}^{H-1} w[i,j]}} \tag{B-68}$$

$$w[i,j] = \left|\sin(\pi\frac{i}{W})\cdot\sin(\pi\frac{j}{H})\right| \tag{B-69}$$

The luminance indicator is then calculated by averaging the frame-wise edginess distortions over time:

$$LumIndicator = \frac{1}{N}\sum_{t=0}^{N-1} e_Y[t] \tag{B-70}$$

B.1.10.3 Chrominance indicator

The chrominance indicator uses a similar approach as that used for the luminance indicator.

The HVS is less sensitive to errors in the colour difference components that occur in areas that have saturated colours. This is even emphasized in bright areas. To reflect this, the colour saturation is computed as follows. As for the deviation signal, the maximum of the colour saturation of the reference signal and the degraded signal is taken:

$$Mx[i,j,t] = \sqrt{\left(S_{A,cb}[i,j,t]-128\right)^2 + \left(S_{A,cr}[i,j,t]-128\right)^2} \tag{B-71}$$

$$My[i,j,t] = \sqrt{\left(P_{A,cb}[i,j,t]-128\right)^2 + \left(P_{A,cr}[i,j,t]-128\right)^2} \tag{B-72}$$

$$dev_{cb;cr}[i,j,t] = \max\left(Mx[i,j,t];My[i,j,t]\right) \tag{B-73}$$

Then the normalized change in edginess of both colour components are evaluated:

$$e_{cb}[i,j,t] = \frac{P_{edge,cb}[i,j,t] - S_{edge,cr}[i,j,t]}{S_{edge,cb}[i,j,t] + 40 + 0.8\cdot dev[i,j,t]} \tag{B-74}$$

$$e_{cr}[i,j,t] = \frac{P_{edge,cr}[i,j,t] - S_{edge,cr}[i,j,t]}{S_{edge,cr}[i,j,t] + 40 + 0.8\cdot dev_{cb;cr}[i,j,t]} \tag{B-75}$$

And clipped locally to the range [–40,40].

$$e_{clipped,cb}[i,j,t] = \begin{cases} -40 & if \quad e_{cb}[i,j,t] \le -40 \\ 40 & if \quad e_{cb}[i,j,t] \ge 40 \\ e_{cb}[i,j,t] & otherwise \end{cases} \tag{B-76}$$

$$e_{clipped,cr}[i,j,t] = \begin{cases} -40 & if \quad e_{cr}[i,j,t] \le -40 \\ 40 & if \quad e_{cr}[i,j,t] \ge 40 \\ e_{cr}[i,j,t] & otherwise \end{cases} \tag{B-77}$$

The aggregation over space is done using a weighted sum of the clipped change in edginess image:

$$e_{cb}[t] = \frac{\sum_{i=0}^{W-1}\sum_{j=0}^{H-1} e_{clipped,cb}[i,j,t]w[i,j]}{\sum_{i=0}^{W-1}\sum_{j=0}^{H-1} w[i,j]} \tag{B-78}$$

$$e_{cr}[t]=\frac{\sum\limits_{i=0}^{W-1}\sum\limits_{j=0}^{H-1} e_{clipped,cr}[i,j,t]w[i,j]}{\sum\limits_{i=0}^{W-1}\sum\limits_{j=0}^{H-1} w[i,j]} \tag{B-79}$$

$$w[i,j]=\left|\sin(\pi\frac{i}{W})\cdot\sin(\pi\frac{j}{H})\right| \tag{B-80}$$

The chrominance indicator is then calculated by averaging the frame-wise edginess distortions over time:

$$ChromIndicator=\frac{0.5}{N}\sum_{t=0}^{N-1}\left(e_{cb}[t]+e_{cb}[t]\right) \tag{B-81}$$

B.1.10.4 Temporal variability indicators

The edginess indicator is a pure spatial indicator. However, the spatial content of a sequence is judged more critically in case of still images than for images with fast motion and rapid changes. To reflect this, the positive contributions to the $DMOS_P$ from *e* (and *n*) indicator should be compensated by a negative contribution from an indicator that measures the temporal variability of the source or processed video sequence. The (peak) temporal variability of the processed video signal is also influenced by transmission errors and the presence or absence of frame repeats. As a result, the temporal variability is best measured on the luminance of the source sequence:

$$d[i,j,t]=\left|S_{A,Y}[i,j,t]-S_{A,Y}[i,j,t-1]\right|-\left|P_{A,Y}[i,j,t]-P_{A,Y}[i,j,t-1]\right| \tag{B-82}$$

The HVS reacts differently if a new component is introduced to the signal than if it is removed from the signal. Therefore, two different indicators are evaluated. To measure the omitted component part, the negative part of the variability measure is taken into account:

$$d_{omitted}[i,j,t]=\begin{cases}0 & \mathit{if}\ d[i,j,t]\le 0\\ -d[i,j,t] & \mathit{otherwise}\end{cases} \tag{B-83}$$

The aggregation over space is done using an L norm of the omitted component:

$$d_{omitted}[t]=\frac{1}{H\cdot W}\sum_{i=0}^{H-1}\sum_{j=0}^{W-1} d_{omitted}[i,j,t] \tag{B-84}$$

The omitted component indicator is then calculated by averaging the frame-wise omitted distortions over time:

$$OmittedComponentIndicator=\frac{1}{N}\sum_{t=0}^{N-1} d_{omitted}[t] \tag{B-85}$$

The positive part of the variability measure is taken to measure the introduced change component indicator:

$$d_{introduced}[i,j,t]=\begin{cases}0 & \mathit{if}\ d[i,j,t]\ge 0\\ d[i,j,t] & \mathit{otherwise}\end{cases} \tag{B-86}$$

The aggregation over space is done using an L5 norm of the introduced component:

$$d_{introduced}[t] = \sqrt[5]{\frac{1}{W \cdot H} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} d_{introduced}^{5}[i,j,t]} \tag{B-87}$$

The introduced component indicator is then calculated using an L2 norm over the frame wise introduced distortions over time:

$$IntroducedComponentIndicator = \sqrt{\frac{1}{N} \sum_{t=0}^{N-1} d_{introduced}^{2}[t]} \tag{B-88}$$

B.1.11 Temporal distortion analysis

When a distorted sequence contains exactly the same image twice, it is said to be repeated. In PEVQ, the analysis of the frame repeats is done based on the information stored in the unsmoothed matching vector: *MatchingPictureIndex[t]* calculated as described in clause B.1.7.1.

B.1.11.1 Definition of sorted weight sum

Let L_{sort} be the descended sorted representative of an input vector L of length N.

Then the return value of the function:

$$SWSum(L, w) = \sum_{i=0}^{N-1} L_{sort}[i] \cdot \min\left(0.125, w^{-i}\right) \tag{B-89}$$

is defined as the sorted weighted Sum of *L* with the weight *w*.

B.1.11.2 Definition of MOS degradation function

The MOS degradation function is defined as:

$$MosDeg\left(n_{fr}\right) = \begin{cases} 0.0562 \cdot \left(\ln\left(25 n_{fr}\right)\right)^{2} + 0.1918 \cdot \ln\left(25 n_{fr}\right) + 0.2548 & if \quad n_{fr} > 0.04 \\ 0 & else \end{cases} \tag{B-90}$$

B.1.11.3 Analysis of frame repeats and frame skips

In a first step, the anomalous frame segments are extracted and the start frame of each segment, the length of each segment and the numbers of skipped frames in each segment are stored in a *MatchInfo* structure according to following pseudo code:

```
Int Repeat=1
Int Nr    =0

for(i=StartMatchIndex+1;i<StopMatchIndex;i++)
     {
          if(MatchingPictureIndex[i+1]== MatchingPictureIndex[i])
          {
                    Repeat++
          }
          else
          {
               MatchInfo[Nr].StartFrame=max(i-Repeat+1,0)
               MatchInfo[Nr].RepeatFrame=max(Repeat,0)
               MatchInfo[Nr].iSkipFrame=max(MatchingVector[i+1]-
                                                    MatchingVector[i]-1,0)
               Repeat=1
               Nr++
          }
     }
```

A histogram $U_{fr,}$ which contains the number of frame repetitions along the first coordinate and the frame skips along the second coordinate, is created:

$$U_{fr}\left[n_{fr},n_{fs}\right]=\sum_{i=1}^{N} v\left[i,n_{fr},n_{fs}\right]$$

with

$$v\left[i,n_{fr},n_{fs}\right]=\begin{cases}1 & if \quad \left\{\begin{array}{l} MatchInfo[i].repeatFrame = n_{fr} \cap \\ MatchInfo[i].SkipFrame\cdot = n_{fs}\end{array}\right. \\ 0 & else\end{cases} \qquad \text{(B-91)}$$

Next, all frame repeat/frame skip combinations which appear less than 3 times are eliminated:

$$\tilde{U}_{f}\left[n_{fr},n_{fs}\right]=\begin{cases}U_{fr}\left[n_{fr},n_{fs}\right] & if \quad U_{fr}\left[n_{fr},n_{fs}\right]>3 \\ 0 & else\end{cases} \qquad \text{(B-92)}$$

And then all frame repeat occurrences which do not exceed 15% of the signal length *Nf* are set to 0:

$$U_{f}\left[n_{fr},n_{fs}\right]=\begin{cases}\tilde{U}_{fr}\left[n_{fr},n_{fs}\right] & if \quad \sum_{i=0}^{N_d} \tilde{U}_{f}\left[n_{fr},n_{fs}\right]>\left\lceil\frac{0.15\cdot Nf}{n_{fr}}\right\rceil \\ 0 & else\end{cases} \qquad \text{(B-93)}$$

The histogram elements which are 0 are used in the evaluation of the anomalous frame repetitions, e.g., those which occur as isolated one-time events.

The first 500 ms and the last 500 ms of the signals are not taken into account because the subjects are distracted by switching from or to the grey background.

In the next step, the elements of interest of *MatchInfo* are selected and copied to a new structure *AnomalousFrameRep*:

```
Count=0
For(i=0;i<N-1;i++)
{
     If((MatchInfo[i].StartFrame>0.5*Fps) &&
            (MatchInfo[i].StartFrame<Nf*Fps-0.5*Fps) &&
            (Uf[MatchInfo[i].RepeatFrame, MatchInfo[i].SkipFrame]==0))
     {
            AnomalousFrameRep[Count]=MatchInfo[i]
            Count++
     }
}
```

B.1.11.4 Detection of related temporal degradations

It should be noted that the distance between two distortions is important. Several freeze conditions with a short distance in between are rated more like one large distortion, and they are usually preferred to several, over the sequence distributed distortions.

In the following, frame freezes with a short distance in between are refered to as being in a connected condition.

A connected condition is assumed if the distance is shorter than the combined number of frames distorted in the current distortion. The connected distortions lead to a combined degradation and are summarized prior to the sequence distortion calculation.

The following pseudo code assigns the flag *ConnectFlag* when the distortions are connected:

```
ConnectLength=0
ConnectRange=0

For(i=0;i<N-1;i++)
{
     ConnectLength += AnomalousFrameRep[i].RepeatFrame

     ConnectRange=AnomalousFrameRep[i].StartFrame+
                          AnomalousFrameRep[i].RepeatFrame+
                         Min(ConnectLength,floor(Fps))
     If(AnomalousFrameRep[i+1].StartFrame<ConnectRange)
     (
               AnomalousFrameRep[i].ConnectFlag=true
     }
     Else
     {
          AnomalousFrameRep[i].ConnectFlag=false
          ConnectLength=0
     }
}
AnomalousFrameRep[N-1].ConnectFlag=false
```

B.1.11.5 Array of temporal degradations

The array of temporal degradation stores the individual MOS values for each individual distortion:

```
Int Count=0
Int NrCont=0
     For(i=0;i<N;i++)
     {
          RepScore=MosDeg(AnomalousFrameRep[i].RepeatFrame/Fps)
          SkipScore=MosDeg(AnomalousFrameRep[i].SkipFrame/Fps)

          Continuous[NrCont]=0.7981483*RepScore+
                                            0.2018517*SkipScore
          NrCont++

          If(AnomalousFrameRep[i].Connected==false)


          {
               ArrayOfDegradations[Count]=SWSum(Continuous,1.9515)
               Count++
               NrCont=0
          }
     }
```

B.1.11.6 Frame repeat indicator

Then the *FrameRepeatIndicator* is given by:

$$FrameRepeatIndicator = SWSum\left(ArrayOfDegradation, 2.2288\right) \qquad \text{(B-94)}$$

B.1.12 Aggregation of indicators

The perceived video quality is estimated by mapping the indicators to a single number using a sigmoid approach. Let $I[i]$ represent the indicators.

Then the mapping function may be defined by a set of input scaling factors $I_{\min}[i]$, $I_{\max}[i]$, a set of scaling factors $w_x[i]$, and a set of output scaling factors:

$$I_{\lim}[i] = \begin{cases} I_{\min}[i] & if \quad I[i] < I_{\min}[i] \\ I_{\max}[i] & if \quad I[i] > I_{\max}[i] \\ I\,[i] & otherwise \end{cases} \tag{B-95}$$

$$Score = LinearOffset + \sum_{i=0}^{4} \frac{w[i]}{1 + e^{\alpha[i] \cdot I_{\lim}[i] + \beta[i]}} \tag{B-96}$$

$$OMOS = \begin{cases} 1 & if \quad Score < 1 \\ 5.5 & if \quad Score > 5.5 \\ Score & \quad otherwise \end{cases} \tag{B-97}$$

Different mappings were used for VGA, CIF and QCIF resolution, presented in the tables below.

Mapping coefficients used for VGA resolution:

Index (i)	**Indicator ($I[i]$)**	**$I_{\min}[i]$**	**$I_{\max}[i]$**	**w[i]**	**α[i]**	**β[i]**
0	*LumIndicator*	0.0000000	26.3458920	5.5178358	0.1982675	–1.9184154
1	*ChromIndicator*	0.0888870	11.9341383	–61.9967023	0.8956342	–14.587778
2	*OmittedComponent Indicator*	0.0000000	1603.352661	–12.8507869	0.0026048	2.3705606
3	*IntroducedComponent Indicator*	0.0000000	44.0389137	–0.2219432	0.7256163	–15.768180
4	*FrameRepeatIndicator*	0.0000000	3.3093989	27700.040463	2.4068676	11.2761009
LinearOffset		63.1413711				

Mapping coefficients used for CIF resolution:

Index (i)	**Indicator ($I[i]$)**	**$I_{\min}[i]$**	**$I_{\max}[i]$**	**w[i]**	**α[i]**	**β[i]**
0	*LumIndicator*	0.000000	26.3458919	5.3172446	0.18924704	–1.8354605
1	*ChromIndicator*	0.088887	11.9341383	492.6884156	0.56603720	7.2721399
2	*OmittedComponent Indicator*	0.000000	1603.35266	–26163.42844	0.00234274	10.066421
3	*IntroducedComponent Indicator*	0.000000	44.0389137	–1.2873113	0.04960221	–0.5178041
4	*FrameRepeatIndicator*	0.000000	3.3093988	0.9729061	5.57149036	0.7472324
LinearOffset		1.6945190				

Mapping coefficients used for QCIF resolution:

Index (i)	**Indicator ($I[i]$)**	$I_{min}[i]$	$I_{max}[i]$	$w[i]$	$\alpha[i]$	$\beta[i]$
0	*LumIndicator*	0.0000000	26.3458920	6.1842156	0.1683161	–1.4493381
1	*ChromIndicator*	0.0888870	11.9341383	2.5008131	1.3506481	–17.748015
2	*OmittedComponent Indicator*	0.0000000	1603.35266	–5699.585091	0.0036791	8.7293193
3	*IntroducedComponent Indicator*	0.0000000	44.0389137	–0.8661656	0.3119277	–6.7674783
4	*FrameRepeatIndicator*	0.0000000	3.3093989	0.2176915	6.0962224	–2.4768872
LinearOffset		–0.9269844				

Bibliography

[B1] A.P. Hekstra, *et al.*, *PVQM – A perceptual video quality measure*, in Signal Processing: Image Comminicatins 2002, vol. 17, pp 781-798, Elsevier.

[B2] Al Bovik, Ed., *Handbook of Image and Video Processing*, Elsevier Academic Press, second edition, 2005.

[B3] Marcus Bakowsky, Jens Bialkowski, Roland Bitto, André Kaup, *Temporal registration using 3D phase correlation and a maximum likelihood approach in the perceptual evaluation of video quality*. Proc. IEEE International Workshop on Multimedia Signal Processing, 2007, pp 195-198.

[B4] Marcus Bakowsky, Roland Bitto, Jens Bialkowski, André Kaup, *Comparison of matching strategies for temporal frame registration in the preceptual evaluation of video quality*. Proc. of the Second International Workshop on Video Processing and Quality Metrics for consumer Electronics, 2006.

[B5] Marcus Bakowsky, Eskofier Björn, Jens Bialkowski, André Kaup, *Influence of the Presentation Time on Subjective Votings on Coded Still Images.* ICIP 2006, pp 429-432.

[B6] Marcus Bakowsky, Eskofier Björn, Roland Bitto, Jens Bialkowski, André Kaup, *Perceptually motivated spatial and temporal integration of pixel based video quality measures.* MobConQoE/Qshine, 2007.

[B7] Recommendation ITU-R BT.601 (2007), *Studio encoding parameters of digital television for standard 4:3 and wide screen aspect ratios*.

Annex C

Psytechnics full reference method

(This annex forms an integral part of this Recommendation)

C.1 Scope

This annex provides the normative description of the Psytechnics full reference video quality assessment algorithm. Models conforming to this annex must implement all processes and steps described herein.

C.2 Introduction

The Psytechnics full reference video quality assessment algorithm accommodates human spatial and temporal frequency responses, and contrast sensitivity masking. The identification of perceptually relevant boundaries and the inclusion of a model of the human visual system allow the model to identify and quantify errors perceived by human viewers.

A calibration process was applied using an extensive subjective database of more than 100 000 votes to map the outputs of the model to actual subjective scores. As a result, the Psytechnics video quality assessment model produces (objective) quality predictions that correlate highly with human (subjective) quality judgement. It provides the industry with a reliable alternative tool to time-consuming and costly subjective testing to measure end-user quality of experience (QoE).

C.3 Overview of the Psytechnics full reference video model

An overview of the Psytechnics full reference video quality assessment model is provided in Figure C.1 The model includes 3 main stages: 1) video registration, 2) detection of perceptual features and 3) integration of these features into an overall quality prediction score.

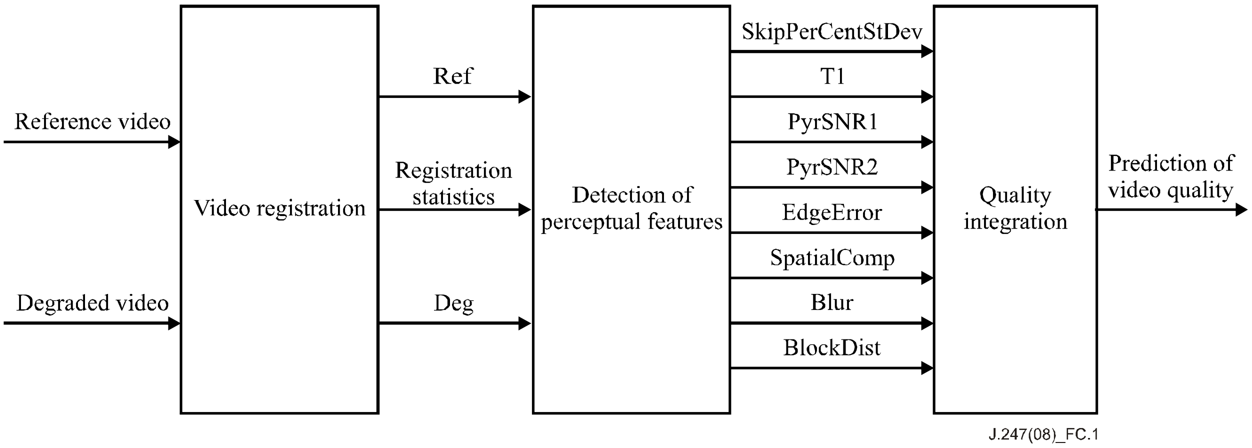


Figure C.1 – Overview of Psytechnics full reference video quality assessment model

The first stage consists of a video registration which temporally pairs each frame in the degraded video to the best matching frame in the reference video. Video registration is necessary to provide accurate quality measurement. From the original reference and degraded videos, matched reference and matched degraded video sequences are obtained, together with registration information. The matched reference video contains all the reference frames that have been matched to frames of the degraded video. The matched degraded video contains all degraded frames for which a reference frame has been identified. The video registration algorithm is able to cope with time-varying temporal and spatial offsets between reference and degraded videos.

The second stage consists of the analysis of a set of perceptually meaningful features extracted either directly from the degraded video or from a comparison between the reference and degraded videos. These perceptual features are not based on any assumption about the type of content in the video or how the video has been encoded and transmitted to the end user. The perceptual features act as a sensory layer that filters out the image components to which the human viewer is insensitive. This is because video codecs reduce the amount of data to encode by applying image processing techniques to remove components of the video signal to which viewers would be least likely to notice as missing. By incorporating a model of the human visual system, this stage of the algorithm is able to determine how effective these processes are and identify visible errors. Where a coding scheme does not successfully achieve this goal, the degradations will be visible to the end user and will therefore form a part of the output from the sensory layer.

In the final stage, the perceptual parameters are combined to produce a single overall prediction of video quality. The optimum form of the integration function was obtained by exercising the model over an extensive set of subjective experiments (training set) and verifying its performance on a set of unknown experiments (validation set).

The format of the input reference and degraded videos supported by the software implementation of the model that was submitted to the VQEG multimedia phase I evaluation was uncompressed AVI with UYVY (YUV 4:2:2) colour space format, as specified by the VQEG multimedia test plan [C1]. However, the quality assessment model is independent from this format and is therefore equally applicable with other formats (e.g., uncompressed AVI RGB24) provided that a proper input filter (e.g., colour space conversion filter or file reader) is applied first.

Both reference and degraded videos must be in progressive format. The absolute frame rate of the degraded video must be identical to the one of the reference video (e.g., 25 fps), where absolute frame rate is the number of (progressive) frames per second. However, the effective frame rate of the degraded video can be different from the frame rate of the reference video, where effective frame rate is the (average) number of unique frames per second. The effective frame rate of the degraded video can also be variable in time. The following example is provided as illustration. Reference video (A) has an absolute frame rate of 25 fps and is encoded with an effective (target) frame rate of 12.5 fps (B). The encoded video is played back (i.e., decoded) and captured at 25 fps (C). Consequently every other frame in C is identical to the previous one[3]. The input reference and degraded videos to the quality assessment model are, respectively, A and C.

C.4 Video registration

Before any quality analysis can be performed, reference and degraded video signals must be temporally and spatially aligned. The Psytechnics full reference video quality assessment model provides an automatic registration capability to align the reference and degraded video sequences. The registration algorithm is designed to handle time-varying temporal misalignment and spatial offset.

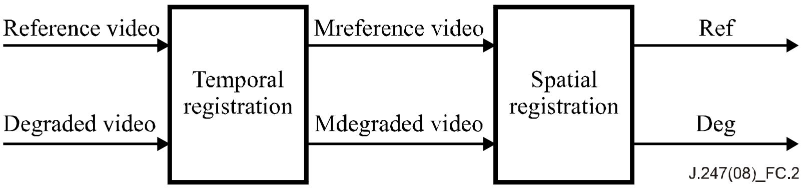


Figure C.2 – Overview of video registration

[3] If the encoding or playback has not introduced additional frame freezes.

C.4.1 Temporal registration

For each frame in the degraded video sequence, the best matching frame is identified in the reference video sequence. The video registration algorithm is computationally much faster than traditional matching techniques (e.g., block matching) and handles time-varying temporal misalignment.

While the VQEG multimedia test [C.1] only evaluated the performance of the objective models using processed videos subject to temporal misalignment within [–0.25, 3] seconds, the video registration algorithm is able to cope with higher misalignment values. There is no theoretical upper or lower limit in the temporal misalignment the algorithm can handle.

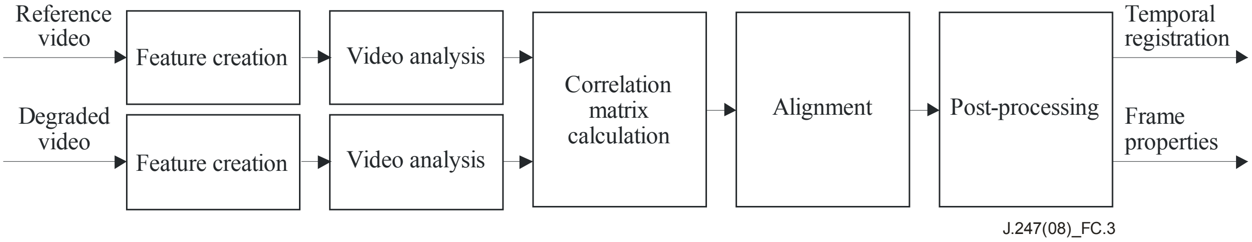


Figure C.3 – Temporal video registration

The temporal registration process can be divided into five steps:

1) Feature creation
2) Video analysis
3) Correlation matrix calculation
4) Alignment
5) Post-processing

C.4.1.1 Feature creation

The feature creation step consists of two sub-steps: sub-sampling and flattening.

The sub-sampling is performed by averaging blocks of pixels as described in the following equation for each of the Y, U and V components:

$$b_{k,l} = \frac{1}{AB} \sum_{x=1}^{A} \sum_{y=1}^{B} p_{(k-1)A+x,(l-1)B+y}$$

where A and B represent the number of pixels per row and column within a block, $p_{x,y}$ the value (Y, U or V) of the pixel at position (x,y) in the image, and $b_{k,l}$ is the resulting sub-sampled value at block position (k,l).

The sub-sampled frames are then flattened into single-dimensional vectors by concatenating each row. Three vectors are generated per frame, one for each YUV component:

$$\begin{cases} \boldsymbol{y} = \begin{bmatrix} b_{1,1} & \cdots & b_{C,1} & b_{1,2} & \cdots & b_{i,j} & \cdots & b_{C,D} \end{bmatrix} \\ \boldsymbol{u} = \begin{bmatrix} b_{1,1} & \cdots & b_{C,1} & b_{1,2} & \cdots & b_{i,j} & \cdots & b_{C,D} \end{bmatrix} \\ \boldsymbol{v} = \begin{bmatrix} b_{1,1} & \cdots & b_{C,1} & b_{1,2} & \cdots & b_{i,j} & \cdots & b_{C,D} \end{bmatrix} \end{cases}$$

where, for each of the vectors **y u v**, $b_{k,l}$ represents the value of the sub-sampled frame at block position (k,l) for the component Y, U and V, respectively. C and D are the number of blocks per row and blocks per column in the sub-sampled frame.

The values of *A*, *B*, *C* and *D* as a function of video resolution are provided in the following table:

Table C.1 – Feature creation parameter values

Resolution	*A*	*B*	*C*	*D*
QCIF	4	4	44	36
CIF	8	8	44	36
VGA	8	8	80	60

C.4.1.2 Video analysis

All frames in the reference and degraded signals are evaluated to determine if they are blank or static.

A frame is marked as static if its correlation with the preceding frame is greater than 0.999. If the previous frame is also marked as static, then for the current frame to be marked as static, the correlation between the current frame and the first frame in the current sequence of consecutive static frames must also be greater than 0.999. This second check prevents a sequence of very slow motion video from being completed marked as static.

A blank frame is defined as being a frame of uniform colour (for example entirely white, blue or black frames are blank frames). A frame is marked as blank if the average standard deviation of the three components (Y,U,V) for the given frame is lower than a threshold of 1000/*N*, where *N* is the number of blocks per frame.

C.4.1.3 Correlation matrix calculation

A correlation matrix **R** is generated from the flattened frames such that each frame from the degraded signal is compared with each frame of the reference signal. To compare the frames, a correlation coefficient is used as described in the following equation:

$$\rho(\boldsymbol{x},\boldsymbol{y}) = \frac{\operatorname{cov}(\boldsymbol{x},\boldsymbol{y})}{\sigma(\boldsymbol{x})\,\sigma(\boldsymbol{y})} = \frac{\sum_{j=0}^{M-1}(x_j-\bar{x})(y_j-\bar{y})}{\sqrt{\sum_{j=0}^{M-1}(x_j-\bar{x})^2\sum_{j=0}^{M-1}(y_j-\bar{y})^2}}$$

$$\text{where } \bar{x} = \frac{1}{M}\sum_{j=0}^{M-1}x_j \text{ and } \bar{y} = \frac{1}{M}\sum_{j=0}^{M-1}y_j$$

and where $\boldsymbol{x}$ and $\boldsymbol{y}$ are two vectors of *M* elements to correlate, x_j and y_j are the *j*-th element in vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively.

The three Y, U and V correlation values are averaged to obtain a single correlation value for a given reference frame *a* and degraded frame *b* comparison:

$$r_{a,b} = \frac{\rho(\mathbf{yref}_a,\mathbf{ydeg}_b)+\rho(\mathbf{uref}_a,\mathbf{udeg}_b)+\rho(\mathbf{vref}_a,\mathbf{vdeg}_b)}{3},$$

where $\mathbf{yref}_a$, $\mathbf{uref}_a$, $\mathbf{vref}_a$ and $\mathbf{ydeg}_b$, $\mathbf{udeg}_b$, $\mathbf{vdeg}_b$ vectors represent the flattened sub-sampled vectors for reference frame *a* and degraded frame *b*.

The correlation matrix is therefore:

$$\boldsymbol{R} = \begin{bmatrix} r_{1,1} & \cdots & r_{A,1} \\ \vdots & r_{a,b} & \vdots \\ r_{1,B} & \cdots & r_{A,B} \end{bmatrix}$$

where $r_{a,b}$ is the correlation value from the comparison between frame a of the reference signal with frame b of the degraded signal, A and B are the number of frames in the reference and received signals, respectively.

C.4.1.4 Alignment

The position of each degraded frame in the reference signal is held in vector **p**, and thus p_{i} represents the position of the i-th frame of the degraded signal in the reference signal. This so-called matching value for each frame in the degraded signal is initialized to a value of –1 so it may be determined whether a matching value has yet been assigned. Each element of **p** has an associated confidence value that is initialized to zero.

The time-alignment process is carried out by sequentially processing overlapping subsets of frames in the degraded video signal and attempting to identify matching frames in the reference signal. Each subset comprises ten successive non-static frames selected from the degraded signal.

A histogram is generated from each subset such that each bin in the histogram represents a relative delay between the degraded signal and the reference signal. Each bin is initialized to zero and then the histogram is updated as follows: for each of the non-blank degraded frames in the subset, the relative delay of the reference frame having the greatest correlation with the selected degraded frame is determined and the bin corresponding to this relative delay is incremented. The search range covers all possible relative delays between the reference and degraded signals.

Once the histogram has been generated, peaks in the histogram are identified such that a bin is marked as a peak if the value of a preceding bin is lower and if the value of a following bin is lower or equal to the value of the bin under consideration. The value of bins outside the search range is considered to be zero.

A confidence value is then assigned to each peak based on the number of selected frames from the subset that contributed to that peak as shown in the table below:

Table C.2 – Peak confidence value mapping

Peak value	0	1	2	3	4	5	6	7	8	9	10
Confidence	0	0	0	1	2	2	3	3	4	4	5

Each frame in the subset that has contributed to a peak with a peak confidence value exceeding a threshold of three is used to update the vector **p** and the associated stored confidence values as follows:

- If the degraded frame of the current subset does not yet have a matched frame in the reference signal, then the matching value p_{i} is set according to the relative delay derived from the peak under consideration and the associated confidence value is set to the confidence of the peak.
- If the reference frame has already been allocated a matching value, then its current stored confidence value is compared with the confidence value of the peak being considered. If the confidence value of the current peak has the higher value, then the matching value p_{i} is set according to the relative delay derived from the peak and the associated confidence value is set to the confidence of the peak; otherwise the matching and confidence values of the current frame are left unchanged.

The above steps are repeated for successive overlapping subsets of frames from the degraded signal until all of the frames of interest in the degraded signal have been included in a subset.

C.4.1.5 Post-processing

Once the matching and confidence values have been updated for all frames of interest in the degraded signal as described above, the following post-processing steps are carried out to try to fill any unmatched gaps between two matched frames in the degraded signal.

If two matched frames of the degraded signal with positions *a* and *b*, referred to as boundary frames, are separated by one or more unmatched frames and the boundary frames have the same matching value as each other, then the matching value of the unmatched frames is set to the same value as that of the boundary frames. This may be summarized as follows:

$$\text{If } \begin{cases} (p_b - p_a) = b - a \quad \text{and} \\ a < b-1 \quad \text{and} \\ p_a \neq -1 \quad \text{and} \\ p_b \neq -1 \quad \text{and} \\ p_j = -1 \text{ for } j \in [a+1, b-1] \end{cases} \quad \text{then} \quad p_i = p_a + (i - a) \quad \text{for} \quad i \in [a+1, b-1]$$

Where p_i is the position of the frame of the reference signal matching the frame of the degraded signal with position *i*.

If the boundary frames do not have the same delay as each other, then the matching values for the intermediate frames of the degraded signal between positions *a* and *b* are set to the positions of the reference frames matching the degraded signal between positions *a* and *b*, starting with the frame of the degraded signal preceding the frame with position *b* and working backwards until either all of the intermediate degraded frames have been matched or all of the intermediate frames of the reference signal have been used. In the latter case, any of the remaining unmatched intermediate frames of the degraded signal that are marked as static are matched to the frame of the reference signal already matched to the frame with position *a*. This last step addresses frame freezes in the degraded signal. This may be summarized as follows:

$$\text{If } \begin{cases} (p_b - p_a) \neq b - a \quad \text{and} \\ p_a < p_b \quad \text{and} \\ a < b-1 \quad \text{and} \\ p_a \neq -1 \quad \text{and} \\ p_b \neq -1 \quad \text{and} \\ p_j = -1 \text{ for } j \in [a+1, b-1] \end{cases}$$

$$\text{then } \begin{cases} p_i = p_b - (b-i) \qquad \text{for} \quad i \in [\max(b-(p_b - p_a), a+1), b-1] \\ \text{if frame } i \text{ is static then} \quad p_i = p_a \quad \text{for } i \in [a+1, \max(b-(p_b - p_a), a+1)] \end{cases}$$

Where p_i is the position of the frame of the signal matching the frame of the degraded signal with position *i*.

Once the post-processing of the temporal video registration is complete, frozen frames are identified. A frame in the degraded signal is marked as frozen if it has been identified as static but the corresponding frame in the reference has not.

C.4.2 Spatial registration

Reference and degraded videos can be spatially misaligned although spatial shifting in the transmission of digital video is very rare. The VQEG multimedia test [C.1] only evaluated the performance of the objective models using processed videos subject to a potential spatial shift within ±1 pixel. However, the video registration algorithm is able to cope with spatial shift values within ±4 pixels in each direction and handles time-varying offsets. The spatial registration is performed using the luminance information in the reference and degraded video sequences. A block diagram of the spatial registration process is provided in Figure C.4.

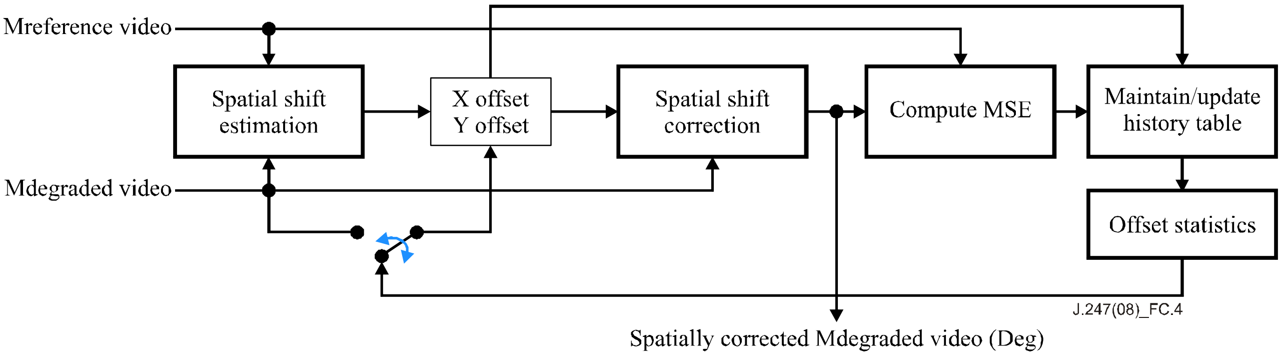


Figure C.4 – Spatial video registration

The spatial offset for each frame is estimated based on information from a history table. Statistics from the history table will decide if the offset needs to be computed or if it can be extracted directly from the table. If a non-zero offset is found between reference and degraded video frames then the degraded frame is shifted back to cancel this spatial misalignment before analysis of the perceptual features.

The history table keeps a record of the (x,y) offsets corresponding to the 10 frames with lowest MSE values. From the history table, a histogram containing *Nbins* bins is computed, where each bin represents one of the possible combinations of (x,y) offsets (see Figure C.5):

for $(x_\mathrm{i}, y_\mathrm{i}, k) = (–XSEARCH, –YSEARCH, 1), ..., (XSEARCH, YSEARCH, Nbins)$:

$$HistXY(\mathrm{k}) = \text{sum of occurrences of } (x_i, y_i)$$

where

$Nbins = (2 * XSEARCH + 1) * (2 * YSEARCH + 1)$

$XSEARCH = YSEARCH = 4$

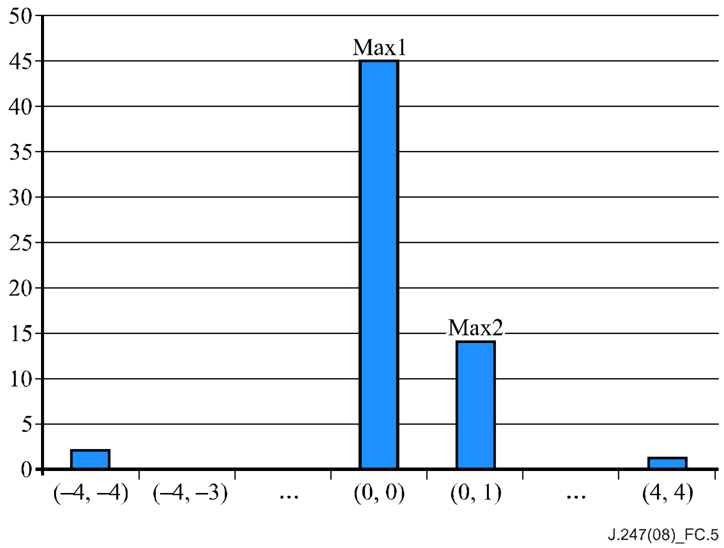


Figure C.5 – Histogram of offsets corresponding to frames with lowest MSE values

The top two number of occurrences *Max*1 and *Max*2 are used to decide which final offset *(Xoffset,Yoffset)* is assigned to the frame:

$Max1 = max(HistXY(k))$

$Max2 = max(HistXY(k) < Max1)$

$if\,(Max1 > 1.5 * Max2)$

$\quad (Xoffset,\ Yoffset) = (x_i,\ y_i) : \{HistXY(k) = Max1\}$

else

Compute offset for the current frame

The spatial offset for the current degraded frame is computed using an *M*N* block in the centre of the degraded frame and looking for the best matching spatial position of this block in the reference[4] frame within a certain spatial search range (see Figure C.6). First, the sum of absolute differences (SAD) between the degraded block and all the corresponding reference blocks within the *(XSEARCH,YSEARCH)* search range is computed:

For $x = -XSEARCH, ..., XSEARCH$ and $y = -YSEARCH, ..., YSEARCH$:

$$SAD(x,y) = \sum_{X=0}^{M-1} \sum_{Y=0}^{N-1} \left| Deg(X + CX, Y + CY) - Ref\left(X + CX + x, Y + CY + y\right) \right|$$

where:

$M = W - 2 * CX$

$N = H - 2 * CY$

$$CX = round\left(\frac{W * 30}{720}\right) and\ CY = round\left(\frac{H * 20}{576}\right)$$

W is the picture width in pixels and *H* is the picture height in pixels.

[4] The temporally matched reference frame.

The values for the different resolutions are:

Table C.3 – Spatial registration parameter values

Resolution	*W*	*H*	*CX*	*CY*
QCIF	176	144	7	5
CIF	352	288	15	10
VGA	640	480	27	17

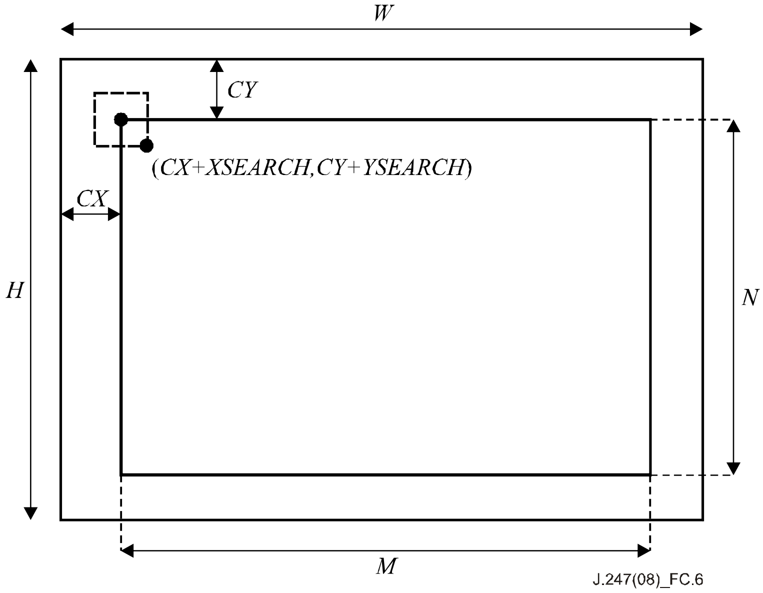


Figure C.6 – Spatial matching

Then the offset for the degraded frame is the one for which the minimum *SAD* value is obtained:

$$(Xoffset, Yoffset) = (x, y) : \{\min(SAD(x, y))\}$$

Finally, the MSE[5] for the current frame is computed. If the MSE of the current frame is lower than the maximum one in the table then the history table is updated with MSE and offset values of the current frame, such that the table always lists the data for the 10 frames with minimum MSE.

C.5 Detection of perceptual features

An overview of the detection and analysis of the perceptual features in the model is provided in Figure C.7. Six analysis modules are used. Five modules focus on the analysis of spatial distortions: spatial frequency, edge distortion, blurring, block distortion and spatial complexity analysis. One module focuses on the analysis of temporal distortions.

Before analysis of the spatial features, both the reference and degraded video sequences are spatially resized to QCIF (176 x 144) resolution using the bilinear interpolation method.

From this point (unless stated otherwise), it is assumed that the terms ‘reference' and ‘degraded' video refer, respectively, to the QCIF-resampled matched reference and matched degraded video sequences, i.e., after temporal and spatial registration.

[5] MSE computed between the reference frame and degraded frame after correction of the spatial shifting.

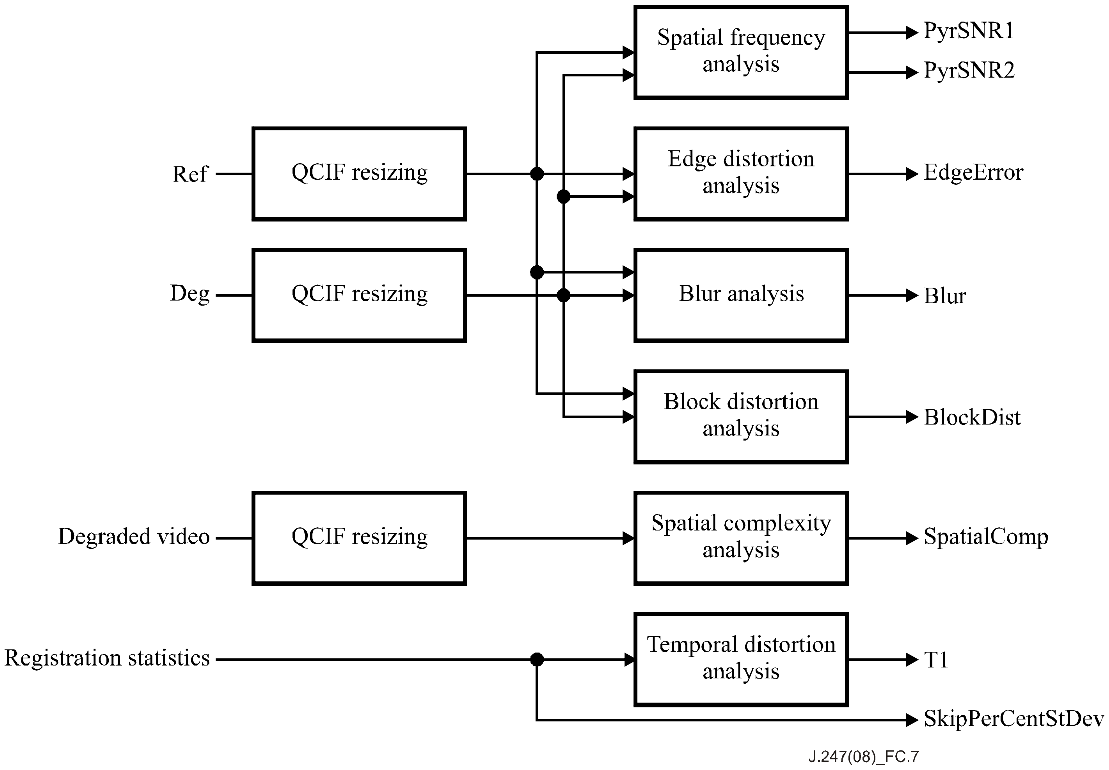


Figure C.7 – Perceptual features analysis

C.5.1 Spatial frequency analysis

The human visual system has a selective sensitivity depending on the spatial orientation and frequency information of the visual stimulus. The quality assessment model therefore includes a spatial frequency analysis that decomposes the video signal into sub-bands and performs an error measurement in some of the sub-bands.

The spatial frequency analysis is achieved by processing the reference and degraded video sequences through 4 steps, as shown in Figure C.8: 1) YUV colour space adaptation, 2) filtering using a contrast sensitivity function, 3) spatial frequency decomposition and 4) error measurement.

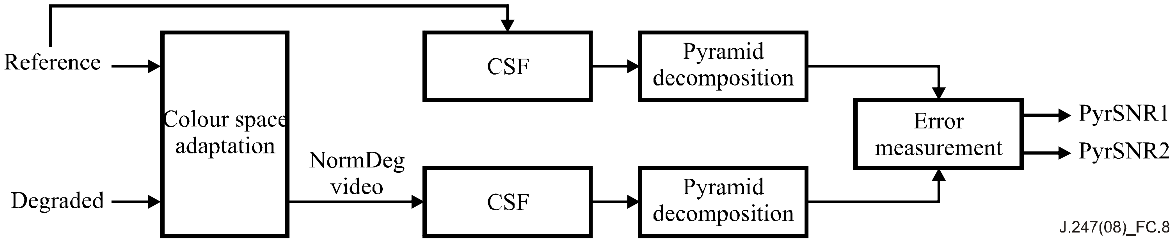


Figure C.8 – Spatial frequency analysis

C.5.1.1 Colour space adaptation

Colour space adaptation is performed on the degraded video sequence: the YUV values of the pixels in the degraded video are modified based on the YUV values of the pixels in the reference video[6]. Colour space adaptation is computed for each pair of matched reference and degraded frames.

[6] The system is, in fact, solved only for the Y component as the subsequent steps in the spatial frequency analysis only use the luminance information in the video.

The transformation vector is computed by solving a linear least squares (LLS) problem using single value decomposition (SVD)[7]:

$$\underset{x}{\text{minimize}} \|b - Ax\| 2$$

for

$$Ax = b$$

⇔

$$\begin{pmatrix} Yd_1^3 & Ud_1^3 & Vd_1^3 & Yd_1^2 & Ud_1^2 & Vd_1^2 & Yd_1 & Ud_1 & Vd_1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ Yd_N^3 & Ud_N^3 & Vd_N^3 & Yd_N^2 & Ud_N^2 & Vd_N^2 & Yd_N & Ud_N & Vd_N & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ \dots \\ x_{10} \end{pmatrix} = \begin{pmatrix} Yr_1 \\ \dots \\ Yr_N \end{pmatrix}$$

where $\|v\|_2$ denotes the L2 norm of vector v, i.e., $\|v\|_2 = \sqrt{v_1^2 + v_2^2 + \ldots + v_N^2}$, $(Yd_k\ Ud_k\ Vd_k)$ are the YUV values of pixel k in the degraded image and Yr_k is the luminance value of pixel k in the reference image.

The transformation vector is computed using pixels of the 44 x 36 sub-resolution image obtained by a linear spatial sampling of the input image in the horizontal and vertical directions, i.e., sub-sampling by a factor of 4 of the 176 x 144 input reference and degraded images.

The luminance value in the degraded frame is then modified using the transformation vector:

$$\begin{pmatrix} Y_{o1} \\ \dots \\ Y_{oN} \end{pmatrix} = \begin{pmatrix} Y_{i1}^3 & U_{i1}^3 & V_{i1}^3 & Y_{i1}^2 & U_{i1}^2 & V_{i1}^2 & Y_{i1} & U_{i1} & V_{i1} & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ Y_{iN}^3 & U_{iN}^3 & V_{iN}^3 & Y_{iN}^2 & U_{iN}^2 & V_{iN}^2 & Y_{iN} & U_{iN} & V_{iN} & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ \dots \\ x_{10} \end{pmatrix}$$

where (Y_{ik}, U_{ik}, V_{ik}) are the YUV values of pixel k in the degraded image before transformation and Y_{ok} is the resulting luminance value of pixel k after adaptation.

Finally, the luminance value is bound within [0,255]:

$$Y_{ok}' = \min(\max(Y_{ok}, 0), 255)$$

C.5.1.2 Contrast sensitivity function

The reference and adapted degraded video sequences are passed to a contrast sensitivity function (CSF) which provides a basic model of the human visual system (HVS) response. The HVS has a limited sensitivity. Contrast sensitivity is generally defined as the inverse of the contrast threshold, which is the minimum contrast necessary for the average viewer to detect a change in intensity. The variations in sensitivity are not linear in the HVS but are a function of the spatial frequencies. The variations as a function of spatial frequency are due to the optics of the human eye combined with the neural circuitry, and these combined effects are referred to as the contrast sensitivity function (CSF).

Our CSF is implemented as a 2D filter applied to the input image using a convolution operation. The convolution kernel is given in clause C.6. The convolution operation is applied such that if the aperture is partially outside the image (i.e., pixels near image borders), the operation interpolates outlier pixel values from the nearest pixels that are inside the image.

[7] The SVD is computed using a divide-and-conquer algorithm [C3], [C4].

C.5.1.3 Pyramid decomposition

Each luminance frame of the reference and degraded videos[8] is processed by a linear multi-scale and multi-orientation image decomposition known as the pyramid transform. A mean squared error measure between the corresponding pyramid arrays is then calculated and the final spatial frequency analysis measure is expressed as the corresponding signal-to-noise ratio value.

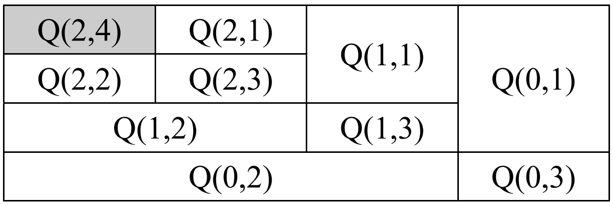
The pyramid decomposition transforms an input image into an array of sub-resolution images as illustrated in Figure C.9. At each stage *s* of the transform, the image is decomposed into 4 quadrants Q(*s*,1), Q(*s*,2), Q(*s*,3) and Q(*s*,4). At the next stage *s*+1, Q(*s*,4) is subsequently decomposed into 4 new quadrants.

Three stages of decomposition are used:

1) Stage 0: The initial image is decomposed into Q(0,1), Q(0,2), Q(0,3) and Q(0,4).

2) Stage 1: Q(0,4) is decomposed into Q(1,1), Q(1,2), Q(1,3) and Q(1,4).

3) Stage 2: Q(1,4) is decomposed into Q(2,1) Q(2,2), Q(2,3) and Q(2,4).



<table>
<tr><td>Q(2,4)</td><td>Q(2,1)</td><td rowspan="2">Q(1,1)</td><td rowspan="3">Q(0,1)</td></tr>
<tr><td>Q(2,2)</td><td>Q(2,3)</td></tr>
<tr><td colspan="2">Q(1,2)</td><td>Q(1,3)</td></tr>
<tr><td colspan="3">Q(0,2)</td><td>Q(0,3)</td></tr>
</table>

Figure C.9 – Image pyramid decomposition

The pixel values in the quadrants at each level of decomposition are obtained using a horizontal and vertical transformation of the input image *Y*, where *Y*(*x,y,i*) is the luminance pixel value at spatial position (*x,y*) for the (reference or degraded) frame at temporal position *i*.

The horizontal transformation divides the input image into 2 halves. The left half contains the average of horizontal pairs of pixels and the right half contains differences between horizontal pairs of pixels:

$$LimitX = \frac{W}{2^{stage+1}}$$

$$LimitY = \frac{H}{2^{stage}}$$

for $x = 0,\ldots, (LimitX–1)$ and $y = 0,\ldots, (LimitY–1)$:

$$H(x,y,i) = 0.5*(Y(2x,y,i)+Y(2x+1,y,i))$$

$$H(x+LimitX,y,i) = Y(2x,y,i)-Y(2x+1,y,i)$$

[8] For the reference video signal, this is the reference video after filtering with the contrast sensitivity function. For the degraded video signal, this is the degraded video after colour space adaptation and filtering with the contrast sensitivity function.

The subsequent vertical transformation divides the image into 2 halves. The top half contains the average of vertical pairs of pixels and the bottom half contains differences between vertical pairs of pixels:

$$LimitX = \frac{W}{2^{stage}}$$

$$LimitY = \frac{H}{2^{stage+1}}$$

for $x = 0,..., (LimitX–1)$ and $y = 0,..., (LimitY–1)$:

$$Pyr(x, y, i) = 0.5 * (H(x, 2y, i) + H(x, 2y+1, i))$$

$$Pyr(x, y + LimitY, i) = H(x, 2y, i) - H(x, 2y+1, i)$$

An error measurement is then computed between the reference and degraded pyramid arrays:

for $stage = 0, 1, 2$ and $quadrant = 1, 2, 3$:

$$E(stage, quadrant, i) = \sum_{y=StartY}^{EndY} \sum_{x=StartX}^{EndX} (RefPyr(x, y, i) - DegPyr(x, y, i))^2$$

$$dMSE(stage, quadrant, i) = \frac{E(stage, quadrant, i)}{SizeXY(stage)}$$

The x and y limits for the different quadrants are calculated as follows[9]:

$if\ quadrant = 1$

$$StartX = \frac{W}{2^{stage+1}}; StartY = 0;$$

$$EndX = 2 * StartX; EndY = \frac{H}{2^{stage+1}};$$

$$MaxMSE = 255 * 2 * 255 * 2;$$

$else\ if\ quadrant = 2$

$$StartX = 0; StartY = \frac{H}{2^{stage+1}};$$

$$EndX = \frac{W}{2^{stage+1}}; EndY = 2 * StartY$$

$$MaxMSE = 255 * 2 * 255 * 2;$$

$else\ if\ quadrant = 3$

$$StartX = \frac{W}{2^{stage+1}}; StartY = \frac{H}{2^{stage+1}}$$

$$EndX = 2 * StartX; EndY = 2 * StartY;$$

$$MaxMSE = 255 * 4 * 255 * 4;$$

[9] The values of the fourth quadrant are in practice not calculated as it will be subsequently decomposed at the following stage.

The corresponding signal-to-noise ratio value is finally obtained as follows:

$if\ dPyrMSE(stage, quadrant, i) > 0$

$$PyrSNR(stage, quadrant, i) = 10 * \log_{10} \frac{MaxMSE}{dMSE(stage, quadrant, i)}$$

else

$$PyrSNR(stage, quadrant, i) = 10 * \log_{10}(MaxMSE * SizeXY(stage))$$

where

$$SizeXY(stage) = \frac{SizeXYPix}{4^{stage+1}}$$

*SizeXYPix = W*H*

where *W* is the picture width in pixels and *H* is the picture height in pixels.

After the three stages of pyramid image decomposition, several image arrays are therefore obtained. Two elements of the pyramid arrays are used in the quality assessment model. One overall value for each element is obtained for the video sequence by temporally averaging the values obtained for all frames:

$$PyrSNR1 = \frac{1}{N} \sum_{i=1}^{N} PyrSNR(s1, q1, i)$$

$$PyrSNR2 = \frac{1}{N} \sum_{i=1}^{N} PyrSNR(s2, q2, i)$$

where *N* is the number of frames in the video sequence.

Different values of decomposition stage ($s1$, $s2$) and quadrant ($q1$, $q2$) are used depending on the original input video resolution as indicated in Table C.4.

Table C.4 – Quadrant and stage values for QCIF, CIF and VGA

Resolution	*s*1	*q*1	*s*2	*q*2
QCIF	2	1	0	1
CIF	1	3	0	2
VGA	2	2	0	2

C.5.2 Edge distortion analysis

Identification of highly relevant perceptual boundaries of the elements forming the visual stimulus constitutes valuable information for the analysis of perceived quality. The human visual system is highly sensitive to visual degradations around high-frequency information such as edges. Therefore a measure of edge distortion is computed in our quality assessment model.

The edge distortion analysis is achieved by passing the reference and degraded video sequences through an edge detection algorithm to build reference and degraded edge maps. An error measure is then computed between the reference and degraded edge maps, as shown in Figure C.10. Edge distortion analysis is computed using the luminance information of the reference and degraded video sequences.

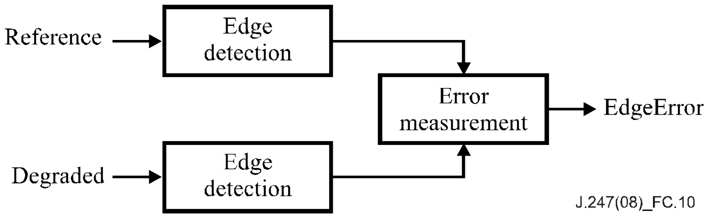


Figure C.10 – Edge distortion analysis

Edge detection in the reference and degraded video sequences is achieved by using the Canny edge detector [C2]. From applying the edge detector on the reference and degraded video sequences, a reference edge map *EMapRef*(*i*) and a degraded edge map *EMapDeg*(*i*) are obtained for each frame *i*:

for each frame $i = 1,...,N$:

$$EMap(x,y,i) = \begin{cases} 1, & if\ pixel(x,y)\ is\ part\ of\ a\ detected\ edge \\ 0, & if\ pixel(x,y)\ is\ not\ part\ of\ a\ detected\ edge \end{cases}$$

Two types of error measure are performed based on the difference between reference and degraded edge maps.

C.5.2.1 Grid error analysis

For each pair of reference-degraded frames, at temporal position *i*, the proportion of edge pixels present in either the reference or the degraded edge map but not present in the other one is computed:

for each frame *i* = 1,..., *N*:

$$EMapDiff(x,y,i) = |EMapRef(x,y,i) - EMapDeg(x,y,i)|$$

$$sum(i) = \sum_{x=X1}^{W-1} \sum_{y=Y1}^{H-1} EMapDiff(x,y,i)$$

$$EEM0RawGrid(i) = \frac{sum(i)}{NPixels}$$

where

$NPixels = (W - X1) * (H - Y1)$

W is the picture width in pixels and *H* is the picture height in pixels

$X1 = 6\ and\ Y1 = 10$.

C.5.2.2 Block-based error analysis

For each pair of reference-degraded frames at temporal position *i*, a block-based analysis is performed using 4x4 non-overlapping blocks. The parameter is computed based on the sum of absolute differences between the reference and degraded edges per block.

First, a measure of the number of edge-marked pixels in each block is calculated, for *Bx* and *By* number of non-overlapping blocks to be analysed in the horizontal and vertical directions, and where *X*1 and *Y*1 define analysis offsets from the frame edge.

for each frame $i = 1,..., N$:

for each block $x = 0,..., Bx - 1$ and $y = 0,..., By - 1$:

$$BlockRef(x, y, i) = \sum_{m=-2}^{1} \sum_{n=-2}^{1} EMapRef(4x + X1 + m, 4y + Y1 + n, i)$$

for each block $x = 0,..., Bx - 1$ and $y = 0,..., By - 1$:

$$BlockDeg(x, y, i) = \sum_{m=-2}^{1} \sum_{n=-2}^{1} EMapDeg(4x + X1 + m, 4y + Y1 + n, i)$$

A measure of difference is calculated for each block:

for $x = 0,..., Bx - 1$ and $y = 0,..., By - 1$:

$$BlockDiff(x, y, i) = \left| BlockRef(x, y, i) - BlockDeg(x, y, i) \right|$$

A measure of difference over the whole frame is calculated from the differences per block:

$$EEM0RawBlockPP3(i) = \frac{1}{BlockSize} \left(\frac{\sum_{x=0}^{Bx} \sum_{y=0}^{By} BlockDiff^{3}(x, y, i)}{NBlocks} \right)^{1/3}$$

where

$$BlockSize = BlockSizeX * BlockSizeY$$

$$Bx = \frac{W - 2 - X1}{BlockSizeX}$$

$$By = \frac{H - 2 - Y1}{BlockSizeY}$$

$$NBlocks = Bx * By$$

$$X1 = 6 \; and \; Y1 = 10$$

$$BlockSizeX = BlockSizeY = 4$$

W is the picture width in pixels and H is the picture height in pixels.

One edge analysis parameter is used in the quality assessment model, either the one based on the grid analysis or the one based on the per-block analysis, depending on the original image resolution:

For QCIF resolution:

$$for \; each \; frame \; i = 1,..., N : EdgeError(i) = EEM0RawBlockPP3(i)$$

For CIF and VGA resolution:

$$for \; each \; frame \; i = 1,..., N : EdgeError(i) = EEM0RawGrid(i)$$

One overall value of the edge distortion metric for the video sequence is obtained by temporally averaging the values obtained for all frames:

$$EdgeError = \frac{1}{N} \sum_{i=1}^{N} edgeError(i)$$

where N is the number of frames in the video sequence.

C.5.3 Block distortion analysis

Most modern coding techniques use block decomposition of the image, motion estimation and pixel value prediction in their compression process. With the transmission of digitally compressed video over error-prone networks, block distortion can occur due to corruption of the data containing information about motion vectors and prediction values. A block distortion parameter is computed using the chrominance information of the reference and degraded video sequences. The analysis is performed using 8x8 non-overlapping blocks according to the following steps:

for each frame i = 1,..., N:

1) For each pair of reference-degraded frames i, decompose each frame into non-overlapping 8x8 blocks and compute the MSE between the corresponding blocks in the V chrominance reference plane *VRef* and degraded plane *VDeg*:

$$\textit{for each block } x = 0,\ldots,Bx-1 \textit{ and } y = 0,\ldots,By-1:$$

$$VBlockMSE(x,y,i) = \frac{1}{BlockSize}\sum_{m=0}^{EndX}\sum_{n=0}^{EndY}\left(VRef(8x+m,8y+n,i) - VDeg(8x+m,8y+n,i)\right)^2$$

where

$$EndX = BlockSizeX - 1,\ EndY = BlockSizeY - 1$$

$$BlockSize = BlockSizeX * BlockSizeY$$

$$Bx = \frac{W}{BlockSizeX}$$

$$By = \frac{H}{BlockSizeY}$$

$BlockSizeX = BlockSizeY = 8$

W is the picture width in pixels and H is the picture height in pixels.

2) Compute the spatial standard deviation of values over the frame i:

$$avVBlockMSE(i) = \frac{1}{NBlocks}\sum_{x=0}^{Bx-1}\sum_{y=0}^{By-1} VBlockMSE(x,y,i)$$

where *Nblocks=Bx * By*

$$VBlockMSEStDev(i) = \sqrt{\frac{\sum_{x=0}^{Bx-1}\sum_{y=0}^{By-1}\left(VBlockMSE(x,y,i) - avVBlockMSE(i)\right)^2}{Nblocks}}$$

The overall value of the block distortion metric for the video sequence is computed using the temporal mean and variation over the sequence:

$$avVBlockMSEStDev = \frac{1}{N}\sum_{i=1}^{N} VBlockMSEStDev(i)$$

$$stdVBlockMSEStDev = \sqrt{\frac{\sum_{i=1}^{N}\left(VBlockMSEStdev(i) - avVBlockMSEStdev\right)^2}{N}}$$

$$BlockDist = \sqrt{avVBlockMSEStdev^2 + stdVBlockMSEStdev^2}$$

where N is the number of frames in the video sequence.

C.5.4 Blur analysis

Blurriness is characterized by a reduced sharpness of edges and spatial detail (attenuation of high spatial frequencies) in the image. It is often caused in compression algorithms by trading off bits to code resolution and motion. A measure of the blur distortion is computed using the luminance information of the reference and degraded video sequences.

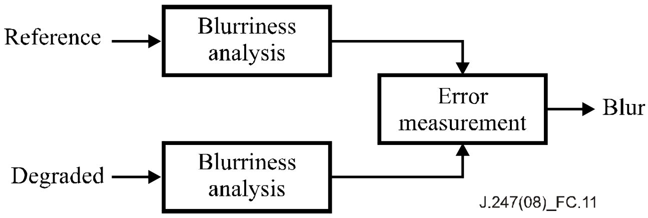


Figure C.11 – Blur distortion analysis

A block diagram of the blur analysis is provided in Figure C.11. First, a measure of blur is computed separately for the pair of reference and degraded frames at temporal position *i*. A blurred image is likely to have all of its high-frequency coefficients set to zero regardless of the content of the image. The blur measure is therefore obtained by counting the number of discrete cosine transform (DCT) coefficients of very small values on each block of 8x8 pixels, and building a histogram with a bin for each DCT coefficient. The bins with small values are then identified and the blur measure is obtained via a weighting depending on the corresponding spatial frequency of the coefficients. The final blur measure used in the quality assessment model is obtained by computing the difference of blur between the reference and degraded frames. The algorithm is described in more detail in the remainder of this clause.

First, a DCT transform is computed on each 8x8 block in the frame *Y*, resulting in 64 DCT coefficients:

```
for each block $x = 0,\ldots, Bx-1$ and $y = 0,\ldots, By-1$:
        $DCTCoeff(k,x,y) = DCT\_TRANSFORM(Y(x,y)), \qquad k = 0,\ldots,NDCTcoeff-1$
```

where

$NDCTcoeff = BlockSizeX * BlockSizeY$

$BlockSizeX = BlockSizeY = 8$

$Y(x,y)$ is either the reference or degraded frame.

The 2D DCT transform is implemented as a 1D DCT on each row followed by a 1D DCT on each column [C5]. For every DCT coefficient, the number of blocks that have this coefficient different from zero is counted. DCT coefficients whose value is inferior to a threshold *DCT_MIN_VALUE* are considered null. This threshold aims at neglecting small values which may result from noise.

```
for each coefficient $k = 0,\ldots,NDCTcoeff - 1$
        $HistCoeff(k) = 0$

for each block $x = 0,\ldots,Bx-1$ and $y - 0,\ldots,By-1$:
        if $|DCTCoeff(k,x,y)| > DCT\_MIN\_VALUE$
            $HistCoeff(k) = HistCoeff(k) + 1$
```

where *DCT_MIN_VALUE* = 8.

The counter value for each DCT coefficient is divided by the number of times that the DC coefficient is different from zero. This results in 64 normalized values for the bins in the histogram, one per DCT coefficient. The value of the first bin, which represents the one for the DC coefficient, is therefore always equal to 1. The right-most bin value indicates the same measure for the AC coefficient of highest frequency.

The number of coefficients that are close to zero in the image is calculated, i.e., the number of values in the histogram that are smaller than a threshold *MIN_TH*. A weighting matrix is applied to give more weight to the DCT coefficients on the central diagonal.

```
$DCTBlur = 0$
    $for\ each\ coefficient\ k = 0,..., NDCTcoeff - 1$:
        $if\ HistCoeff(k) < MIN\_TH$
            $DCTBlur = DCTBlur + weight(k)$
```

where

MIN_TH = 0.1

Weight = { 149, 130, 112, 93, 74, 56, 37, 19,
130, 149, 130, 112, 93, 74, 56, 37,
112, 130, 149, 130, 112, 93, 74, 56,
93, 112, 130, 149, 130, 112, 93, 74,
74, 93, 112, 130, 149, 130, 112, 93,
56, 74, 93, 112, 130, 149, 130, 112,
37, 56, 74, 93, 112, 130, 149, 130,
19, 37, 56, 74, 93, 112, 130, 149}

The blur measure for the frame is then re-scaled to [0,100], with 0 indicating no blur and 100 indicating maximum blur:

$$Bl = round\left(\frac{DCTblur}{64}\right)$$

The DCT-based blur calculation described above is applied separately on the reference and degraded frames, for each pair of reference and degraded frames at temporal position *i.* The blur distortion metric used in the quality assessment model is then computed from the difference between the blur in the reference and degraded frames:

for each frame i = 1,...,*N*:

$$Blur(i) = max(BlDeg(i) - BlRef(i), 0)$$

One overall value of the blur distortion metric for the video sequence is obtained by temporally averaging the values obtained for all frames:

$$Blur = \frac{1}{N}\sum_{i=1}^{N} Blur(i)$$

where *N* is the number of frames in the video sequence.

C.5.5 Spatial complexity analysis

The spatial complexity analysis is computed using the luminance information in the degraded[10] video sequence. For each video frame *i*, a measure of horizontal and vertical complexity is computed based on the proportion of adjacent pixels that have changed in the x and y direction. A low (high) proportion means that the complexity of the image is low (high) and vice versa.

Counters are initialized for each frame *i*:

```
for each frame $i = 1,...,N$:
        $LastDiffX(i) = 0$ and $LastDiffY(i) = 0$
        $SumX(i) = 0$ and $SumY(i) = 0$
```

Changes in x and y directions are calculated as follows:

```
for each frame $i = 1,...,N$:
    for $x = 1,...,W-1$ and $y = 1,...,H-1$:
        $DiffX(x, y, i) = Y(x, y, i) - Y(x-1, y, i)$
        if (($DiffX(x, y, i) > 0$) AND ($LastDiffX(i) < 0$))
            $SumX(i) = SumX(i) + 1$
        if (($DiffX(x, y, i) < 0$) AND ($LastDiffX(i) > 0$))
            $SumX(i) = SumX(i) + 1$
        $LastDiffX(x, y, i) = DiffX(x,y,i)$
        $DiffY(x, y, i) = Y(x, y, i) - Y(x, y-1, i)$
        if (($DiffY(x, y, i) > 0$) AND ($LastDiffY(i) < 0$))
            $SumY(i) = SumY(i) + 1$
        if (($DiffY(x, y, i) < 0$) AND ($LastDiffY(i) > 0$))
            $SumY(i) = SumY(i) + 1$
        $LastDiffY(x, y, i) = DiffY(x,y,i)$
```

where *Y(x,y,i)* is the pixel luminance value at spatial location (*x,y*) in the frame at temporal location *i*.

The spatial complexity for the frame is then obtained as follows:

for each frame i = 1,...,*N*:

$$Ntot = (H-1)*(W-1)$$

$$SpatialComp(i) = 100 * \frac{SumX(i) + SumY(i)}{2 * Ntot}$$

One overall value of the spatial complexity for the video sequence is obtained by temporally averaging the values obtained for all frames:

$$SpatialComp = \frac{1}{N}\sum_{i=1}^{N} SpatialComp(i)$$

[10] Here the term degraded video refers to the original degraded video (and not the matched degraded video) after QCIF resizing.

where *N* is the number of frames in the video sequence.

C.5.6 Temporal distortion analysis

The perceptual impact of temporal distortions in the degraded video sequence is accounted for by two parameters in the model: *SkipPerCentStDev* and *T*1.

From the temporal registration, all dropped frames in the degraded video sequence are identified. A frame *i* is marked by a value *SkipFlag*(*i*)=1 if it is identified as a dropped frame:

$$SkipFlag(i) = \begin{cases} 1, & if\ \ frame\ i\, is\ marked\ as\ a\ dropped\ frame \\ 0, & otherwise \end{cases}$$

SkipPerCentStDev is then computed as follows:

$$avSkipFlag = \frac{1}{N}\sum_{i=1}^{N} SkipFlag(i)$$

$$stdSkipFlag = \sqrt{\frac{\sum_{i=1}^{N}\left(SkipFlag(i) - avSkipFlag\right)^2}{N}}$$

$$SkipPerCen\,tStDev = BoundNorm\,(stdSkipFla\,g, x1, x2, a3, a2, a1, a0)$$

where

$x1 = 0.3307$; $x2 = 4.8754$; $a1 = –0.030818$; $a2 = 0.42321$; $a3 = –1.9498$; $a4 = 5.4698$.

The transformation function y=*BoundNorm*(x,$x1$,$x2$,$a3$,$a2$,$a1$,$a0$) bounds the variable x within a specified range [$x1$,$x2$] and then applies a third-order polynomial function as follows:

$$if\ \ (x < x1) x = x1$$

$$if\ \ (x > x2) x = x2$$

$$y = a3 * x^3 + a2 * x^2 + a1 * x + a0$$

From the temporal registration, all frozen frames in the degraded video sequence are identified. A frame *i* is marked by a value *FreezeFlag(i)*=1 if it is identified as a frozen frame:

$$FreezeFlag(i) = \begin{cases} 1, & if\ \ frame\ i\, is\ marked\ as\ a\ frozen\ frame \\ 0, & otherwise \end{cases}$$

A cumulative histogram representing the distribution of frozen frames in the video is built. In this histogram, each bin *FrDur* represents the duration (ms) of an individual freeze event, where an individual freeze event is identified by a set of consecutive frozen frames. The cumulative value *FrTotDur* for each bin is the total duration (ms) obtained by adding all contributions from freeze events of the same individual duration. The histogram in Figure C.12 shows the example of a video exhibiting several instances of 2 freeze durations, five of 400 ms and one of 800ms.

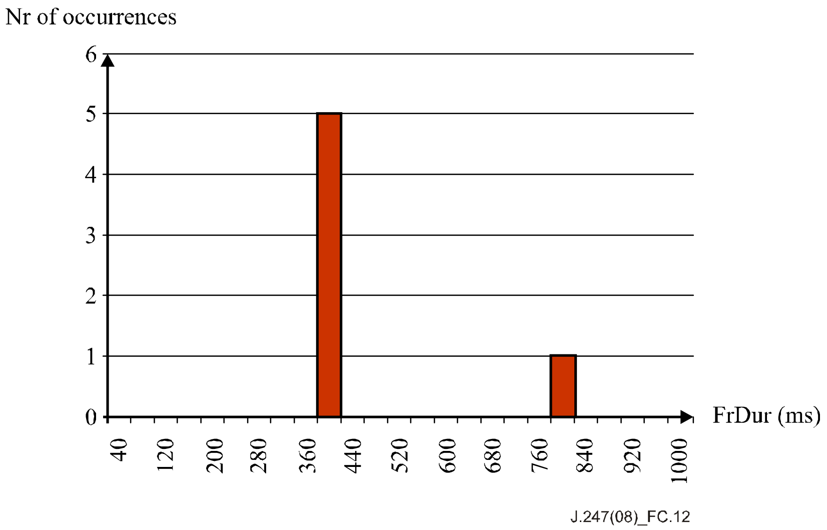


Figure C.12 – Histogram of temporal freezes

For each bin *i* in the histogram of temporal freezes:

$$FrDurPercent_i = \frac{FrDur_i}{TotDur} * 100$$

$$FrTotDurPercent_i = \frac{FrTotDur_i}{TotDur} * 100$$

where *TotDur* is the total duration of the degraded video.

An individual T_i measure is computed for each bin of the histogram according to the following equations:

$$T_i = 1/\left(F2\left(FrTotDurPercent_i\right) * F1\left(FrDurPercent_i\right) + F3\left(FrTotDurPercent_i\right)\right)$$

$$F1(x1) = a1 + b1 * log(c1 * x1 + d1)$$

$$F2(x2) = a2 * x2^2 + b2$$

$$F3(x2) = a3 * x2^2 + b3$$

where

$a1$ = 5.767127; $b1$ = –0.580342; $c1$ = 3.442218; $d1$ = 3.772878

$a2$ = –0.00007; $b2$ = –0.088499

$a3$ = 0.000328; $b3$ = 0.637424

log represents the natural logarithm function.

The value of each T_i is then bound between [1,5]:

$$T_i' = min\left(max\left(T_i, 1\right), 5\right)$$

Finally, the parameter $T1$ is obtained by:

$$T1 = min\left(T_i'\right)$$

C.5.7 Quality integration function

The overall quality prediction $DMOS_p$ is obtained by a linear combination of the perceptual parameters:

$$DMOS_p = K + \sum_{i=1}^{8} P(i) * w(i)$$

The different weight values for each parameter in the integration function of the model are provided in Tables C.5, C.6 and C.7, respectively, for QCIF, CIF and VGA resolution.

A post-processing step is finally applied to bound the prediction values within [1,5]:

$$if\ DMOS_p < 1\, then\, DMOS_p = DMOSp + 1$$

$$DMOS_p = min(max(DMOS_p, 1), 5)$$

The model produces overall video quality prediction scores in the range [1,5], where 1 indicates the lowest quality (maximum perceived impairment) and 5 indicates the highest quality (no perceived impairment). The quality prediction values align with the ACR rating scale.

Table C.5 – Integration parameters for QCIF resolution

I	Parameter *P*(*i*)	Weight *w*(*i*)
1	PyrSNR1	0.359501
2	PyrSNR2	–0.268847
3	EdgeError	–2.14558
4	Blur	–0.0182121
5	BlockDist	–0.000862422
6	SpatialComp	0.0130037
7	T1	0.100673
8	SkipPerCentStDev	0.156619
K	–0.657666	

Table C.6 – Integration parameters for CIF resolution

I	Parameter *P*(*i*)	Weight *w*(*i*)
1	PyrSNR1	0.0865316
2	PyrSNR2	0.0516083
3	EdgeError	–1.87902
4	Blur	–0.00286291
5	BlockDist	–0.00128058
6	SpatialComp	0.0320282
7	T1	0.18545
8	SkipPerCentStDev	0.177194
K	–6.00357	

Table C.7 – Integration parameters for VGA resolution

I	Parameter *P*(*i*)	Weight *w*(*i*)
1	PyrSNR1	0.310165
2	PyrSNR2	–0.210169
3	EdgeError	–5.37116
4	Blur	–0.00312647
5	BlockDist	–0.00358783
6	SpatialComp	0.00749104
7	T1	0.150466
8	SkipPerCentStDev	0.365469
K	–2.70389	

C.6 Kernel for the contrast sensitivity function

CSFKERNEL[41][41] = {

9.9373e-008,9.0998e-008,7.1185e-008,3.8227e-008,-5.8642e-009,-5.9016e-008,-1.2422e-007,-2.083e-007,-3.1291e-007,-4.2561e-007,-5.1986e-007,-5.6502e-007,-5.3861e-007,-4.3251e-007,-2.5156e-007,-9.9862e-009,2.6916e-007,5.5253e-007,7.9858e-007,9.6617e-007,1.0256e-006,9.6617e-007,7.9858e-007,5.5253e-007,2.6916e-007,-9.9862e-009,-2.5156e-007,-4.3251e-007,-5.3861e-007,-5.6502e-007,-5.1986e-007,-4.2561e-007,-3.1291e-007,-2.083e-007,-1.2422e-007,-5.9016e-008,-5.8642e-009,3.8227e-008,7.1185e-008,9.0998e-008,9.9373e-008

9.0998e-008,8.1394e-008,6.4176e-008,4.0672e-008,7.4778e-009,-4.429e-008,-1.2216e-007,-2.2217e-007,-3.2633e-007,-4.1262e-007,-4.682e-007,-4.9216e-007,-4.844e-007,-4.3278e-007,-3.132e-007,-1.0504e-007,1.8932e-007,5.3474e-007,8.6755e-007,1.1112e-006,1.201e-006,1.1112e-006,8.6755e-007,5.3474e-007,1.8932e-007,-1.0504e-007,-3.132e-007,-4.3278e-007,-4.844e-007,-4.9216e-007,-4.682e-007,-4.1262e-007,-3.2633e-007,-2.2217e-007,-1.2216e-007,-4.429e-008,7.4778e-009,4.0672e-008,6.4176e-008,8.1394e-008,9.0998e-008

7.1185e-008,6.4176e-008,5.5327e-008,4.398e-008,1.7578e-008,-3.7546e-008,-1.1824e-007,-2.025e-007,-2.7058e-007,-3.2748e-007,-3.9857e-007,-4.9737e-007,-5.9575e-007,-6.2729e-007,-5.249e-007,-2.6401e-007,1.1877e-007,5.4048e-007,9.0748e-007,1.1498e-006,1.2333e-006,1.1498e-006,9.0748e-007,5.4048e-007,1.1877e-007,-2.6401e-007,-5.249e-007,-6.2729e-007,-5.9575e-007,-4.9737e-007,-3.9857e-007,-3.2748e-007,-2.7058e-007,-2.025e-007,-1.1824e-007,-3.7546e-008,1.7578e-008,4.398e-008,5.5327e-008,6.4176e-008,7.1185e-008

3.8227e-008,4.0672e-008,4.398e-008,3.7589e-008,8.6917e-009,-3.7547e-008,-7.7148e-008,-1.007e-007,-1.4295e-007,-2.6106e-007,-4.7142e-007,-7.0907e-007,-8.5604e-007,-8.165e-007,-5.7271e-007,-1.7959e-007,2.8359e-007,7.5104e-007,1.1672e-006,1.4653e-006,1.5753e-006,1.4653e-006,1.1672e-006,7.5104e-007,2.8359e-007,-1.7959e-007,-5.7271e-007,-8.165e-007,-8.5604e-007,-7.0907e-007,-4.7142e-007,-2.6106e-007,-1.4295e-007,-1.007e-007,-7.7148e-008,-3.7547e-008,8.6917e-009,3.7589e-008,4.398e-008,4.0672e-008,3.8227e-008

-5.8642e-009,7.4778e-009,1.7578e-008,8.6917e-009,-9.2315e-009,-4.0039e-009,2.8742e-008,2.5031e-008,-1.0288e-007,-3.6221e-007,-6.3672e-007,-7.7077e-007,-7.0049e-007,-4.9656e-007,-2.7825e-007,-9.3636e-008,1.0841e-007,3.9216e-007,7.3649e-007,1.0284e-006,1.1432e-006,1.0284e-006,7.3649e-007,3.9216e-007,1.0841e-007,-9.3636e-008,-2.7825e-007,-4.9656e-007,-7.0049e-007,-7.7077e-007,-6.3672e-007,-3.6221e-007,-1.0288e-007,2.5031e-008,2.8742e-008,-4.0039e-009,-9.2315e-009,8.6917e-009,1.7578e-008,7.4778e-009,-5.8642e-009

-5.9016e-008,-4.429e-008,-3.7546e-008,-3.7547e-008,-4.0039e-009,7.6238e-008,1.2239e-007,2.3054e-008,-2.06e-007,-3.8398e-007,-3.4876e-007,-1.6691e-007,-1.0627e-007,-3.6893e-007,-

8.4762e-007,-1.1515e-006,-8.9094e-007,2.1116e-008,1.283e-006,2.3658e-006,2.7903e-006,2.3658e-006,1.283e-006,2.1116e-008,-8.9094e-007,-1.1515e-006,-8.4762e-007,-3.6893e-007,-1.0627e-007,-1.6691e-007,-3.4876e-007,-3.8398e-007,-2.06e-007,2.3054e-008,1.2239e-007,7.6238e-008,-4.0039e-009,-3.7547e-008,-3.7546e-008,-4.429e-008,-5.9016e-008

-1.2422e-007,-1.2216e-007,-1.1824e-007,-7.7148e-008,2.8742e-008,1.2239e-007,7.2448e-008,-8.7992e-008,-1.045e-007,1.8098e-007,4.944e-007,2.8671e-007,-6.981e-007,-2.0559e-006,-2.9525e-006,-2.7365e-006,-1.3925e-006,5.1526e-007,2.2805e-006,3.44e-006,3.8331e-006,3.44e-006,2.2805e-006,5.1526e-007,-1.3925e-006,-2.7365e-006,-2.9525e-006,-2.0559e-006,-6.981e-007,2.8671e-007,4.944e-007,1.8098e-007,-1.045e-007,-8.7992e-008,7.2448e-008,1.2239e-007,2.8742e-008,-7.7148e-008,-1.1824e-007,-1.2216e-007,-1.2422e-007

-2.083e-007,-2.2217e-007,-2.025e-007,-1.007e-007,2.5031e-008,2.3054e-008,-8.7992e-008,1.6457e-008,5.3527e-007,9.9763e-007,5.5881e-007,-9.7982e-007,-2.7102e-006,-3.3725e-006,-2.4842e-006,-6.7969e-007,9.6001e-007,1.8364e-006,2.0997e-006,2.1505e-006,2.1652e-006,2.1505e-006,2.0997e-006,1.8364e-006,9.6001e-007,-6.7969e-007,-2.4842e-006,-3.3725e-006,-2.7102e-006,-9.7982e-007,5.5881e-007,9.9763e-007,5.3527e-007,1.6457e-008,-8.7992e-008,2.3054e-008,2.5031e-008,-1.007e-007,-2.025e-007,-2.2217e-007,-2.083e-007

-3.1291e-007,-3.2633e-007,-2.7058e-007,-1.4295e-007,-1.0288e-007,-2.06e-007,-1.045e-007,5.3527e-007,1.1822e-006,7.1213e-007,-9.5653e-007,-2.1799e-006,-1.3378e-006,1.0827e-006,2.7788e-006,1.9212e-006,-1.025e-006,-3.7044e-006,-4.2083e-006,-3e-006,-2.2026e-006,-3e-006,-4.2083e-006,-3.7044e-006,-1.025e-006,1.9212e-006,2.7788e-006,1.0827e-006,-1.3378e-006,-2.1799e-006,-9.5653e-007,7.1213e-007,1.1822e-006,5.3527e-007,-1.045e-007,-2.06e-007,-1.0288e-007,-1.4295e-007,-2.7058e-007,-3.2633e-007,-3.1291e-007

-4.2561e-007,-4.1262e-007,-3.2748e-007,-2.6106e-007,-3.6221e-007,-3.8398e-007,1.8098e-007,9.9763e-007,7.1213e-007,-8.7173e-007,-1.3154e-006,1.5526e-006,5.5969e-006,5.6418e-006,-9.3668e-007,-1.112e-005,-1.8371e-005,-1.7534e-005,-9.0321e-006,1.0465e-006,5.4512e-006,1.0465e-006,-9.0321e-006,-1.7534e-005,-1.8371e-005,-1.112e-005,-9.3668e-007,5.6418e-006,5.5969e-006,1.5526e-006,-1.3154e-006,-8.7173e-007,7.1213e-007,9.9763e-007,1.8098e-007,-3.8398e-007,-3.6221e-007,-2.6106e-007,-3.2748e-007,-4.1262e-007,-4.2561e-007

-5.1986e-007,-4.682e-007,-3.9857e-007,-4.7142e-007,-6.3672e-007,-3.4876e-007,4.944e-007,5.5881e-007,-9.5653e-007,-1.3154e-006,2.8572e-006,8.3023e-006,6.1144e-006,-7.1228e-006,-2.4227e-005,-3.5416e-005,-3.6893e-005,-3.172e-005,-2.5956e-005,-2.3278e-005,-2.2842e-005,-2.3278e-005,-2.5956e-005,-3.172e-005,-3.6893e-005,-3.5416e-005,-2.4227e-005,-7.1228e-006,6.1144e-006,8.3023e-006,2.8572e-006,-1.3154e-006,-9.5653e-007,5.5881e-007,4.944e-007,-3.4876e-007,-6.3672e-007,-4.7142e-007,-3.9857e-007,-4.682e-007,-5.1986e-007

-5.6502e-007,-4.9216e-007,-4.9737e-007,-7.0907e-007,-7.7077e-007,-1.6691e-007,2.8671e-007,-9.7982e-007,-2.1799e-006,1.5526e-006,8.3023e-006,5.8937e-006,-1.1011e-005,-2.8219e-005,-3.1471e-005,-2.9411e-005,-4.3108e-005,-8.2458e-005,-0.00013804,-0.00018657,-0.00020562,-0.00018657,-0.00013804,-8.2458e-005,-4.3108e-005,-2.9411e-005,-3.1471e-005,-2.8219e-005,-1.1011e-005,5.8937e-006,8.3023e-006,1.5526e-006,-2.1799e-006,-9.7982e-007,2.8671e-007,-1.6691e-007,-7.7077e-007,-7.0907e-007,-4.9737e-007,-4.9216e-007,-5.6502e-007

-5.3861e-007,-4.844e-007,-5.9575e-007,-8.5604e-007,-7.0049e-007,-1.0627e-007,-6.981e-007,-2.7102e-006,-1.3378e-006,5.5969e-006,6.1144e-006,-1.1011e-005,-2.7044e-005,-1.5034e-005,3.2296e-007,-4.5274e-005,-0.00017613,-0.00035221,-0.00050528,-0.00058608,-0.00060511,-0.00058608,-0.00050528,-0.00035221,-0.00017613,-4.5274e-005,3.2296e-007,-1.5034e-005,-2.7044e-005,-1.1011e-005,6.1144e-006,5.5969e-006,-1.3378e-006,-2.7102e-006,-6.981e-007,-1.0627e-007,-7.0049e-007,-8.5604e-007,-5.9575e-007,-4.844e-007,-5.3861e-007

-4.3251e-007,-4.3278e-007,-6.2729e-007,-8.165e-007,-4.9656e-007,-3.6893e-007,-2.0559e-006,-3.3725e-006,1.0827e-006,5.6418e-006,-7.1228e-006,-2.8219e-005,-1.5034e-005,1.5119e-005,-6.1694e-005,-0.00032447,-0.0006975,-0.0010633,-0.0013074,-0.0013456,-0.0013095,-0.0013456,-

0.0013074,-0.0010633,-0.0006975,-0.00032447,-6.1694e-005,1.5119e-005,-1.5034e-005,-2.8219e-005,-7.1228e-006,5.6418e-006,1.0827e-006,-3.3725e-006,-2.0559e-006,-3.6893e-007,-4.9656e-007,-8.165e-007,-6.2729e-007,-4.3278e-007,-4.3251e-007

-2.5156e-007,-3.132e-007,-5.249e-007,-5.7271e-007,-2.7825e-007,-8.4762e-007,-2.9525e-006,-2.4842e-006,2.7788e-006,-9.3668e-007,-2.4227e-005,-3.1471e-005,3.2296e-007,-6.1694e-005,-0.00040371,-0.00094651,-0.0015069,-0.0022031,-0.0029691,-0.0032989,-0.0032838,-0.0032989,-0.0029691,-0.0022031,-0.0015069,-0.00094651,-0.00040371,-6.1694e-005,3.2296e-007,-3.1471e-005,-2.4227e-005,-9.3668e-007,2.7788e-006,-2.4842e-006,-2.9525e-006,-8.4762e-007,-2.7825e-007,-5.7271e-007,-5.249e-007,-3.132e-007,-2.5156e-007

-9.9862e-009,-1.0504e-007,-2.6401e-007,-1.7959e-007,-9.3636e-008,-1.1515e-006,-2.7365e-006,-6.7969e-007,1.9212e-006,-1.112e-005,-3.5416e-005,-2.9411e-005,-4.5274e-005,-0.00032447,-0.00094651,-0.0015385,-0.0021655,-0.0038527,-0.0063652,-0.0076636,-0.0077233,-0.0076636,-0.0063652,-0.0038527,-0.0021655,-0.0015385,-0.00094651,-0.00032447,-4.5274e-005,-2.9411e-005,-3.5416e-005,-1.112e-005,1.9212e-006,-6.7969e-007,-2.7365e-006,-1.1515e-006,-9.3636e-008,-1.7959e-007,-2.6401e-007,-1.0504e-007,-9.9862e-009

2.6916e-007,1.8932e-007,1.1877e-007,2.8359e-007,1.0841e-007,-8.9094e-007,-1.3925e-006,9.6001e-007,-1.025e-006,-1.8371e-005,-3.6893e-005,-4.3108e-005,-0.00017613,-0.0006975,-0.0015069,-0.0021655,-0.0034559,-0.0071944,-0.011522,-0.011877,-0.010584,-0.011877,-0.011522,-0.0071944,-0.0034559,-0.0021655,-0.0015069,-0.0006975,-0.00017613,-4.3108e-005,-3.6893e-005,-1.8371e-005,-1.025e-006,9.6001e-007,-1.3925e-006,-8.9094e-007,1.0841e-007,2.8359e-007,1.1877e-007,1.8932e-007,2.6916e-007

5.5253e-007,5.3474e-007,5.4048e-007,7.5104e-007,3.9216e-007,2.1116e-008,5.1526e-007,1.8364e-006,-3.7044e-006,-1.7534e-005,-3.172e-005,-8.2458e-005,-0.00035221,-0.0010633,-0.0022031,-0.0038527,-0.0071944,-0.01237,-0.012859,-0.0029001,0.0047415,-0.0029001,-0.012859,-0.01237,-0.0071944,-0.0038527,-0.0022031,-0.0010633,-0.00035221,-8.2458e-005,-3.172e-005,-1.7534e-005,-3.7044e-006,1.8364e-006,5.1526e-007,2.1116e-008,3.9216e-007,7.5104e-007,5.4048e-007,5.3474e-007,5.5253e-007

7.9858e-007,8.6755e-007,9.0748e-007,1.1672e-006,7.3649e-007,1.283e-006,2.2805e-006,2.0997e-006,-4.2083e-006,-9.0321e-006,-2.5956e-005,-0.00013804,-0.00050528,-0.0013074,-0.0029691,-0.0063652,-0.011522,-0.012859,0.0034758,0.039857,0.061505,0.039857,0.0034758,-0.012859,-0.011522,-0.0063652,-0.0029691,-0.0013074,-0.00050528,-0.00013804,-2.5956e-005,-9.0321e-006,-4.2083e-006,2.0997e-006,2.2805e-006,1.283e-006,7.3649e-007,1.1672e-006,9.0748e-007,8.6755e-007,7.9858e-007

9.6617e-007,1.1112e-006,1.1498e-006,1.4653e-006,1.0284e-006,2.3658e-006,3.44e-006,2.1505e-006,-3e-006,1.0465e-006,-2.3278e-005,-0.00018657,-0.00058608,-0.0013456,-0.0032989,-0.0076636,-0.011877,-0.0029001,0.039857,0.11133,0.15041,0.11133,0.039857,-0.0029001,-0.011877,-0.0076636,-0.0032989,-0.0013456,-0.00058608,-0.00018657,-2.3278e-005,1.0465e-006,-3e-006,2.1505e-006,3.44e-006,2.3658e-006,1.0284e-006,1.4653e-006,1.1498e-006,1.1112e-006,9.6617e-007

1.0256e-006,1.201e-006,1.2333e-006,1.5753e-006,1.1432e-006,2.7903e-006,3.8331e-006,2.1652e-006,-2.2026e-006,5.4512e-006,-2.2842e-005,-0.00020562,-0.00060511,-0.0013095,-0.0032838,-0.0077233,-0.010584,0.0047415,0.061505,0.15041,0.19791,0.15041,0.061505,0.0047415,-0.010584,-0.0077233,-0.0032838,-0.0013095,-0.00060511,-0.00020562,-2.2842e-005,5.4512e-006,-2.2026e-006,2.1652e-006,3.8331e-006,2.7903e-006,1.1432e-006,1.5753e-006,1.2333e-006,1.201e-006,1.0256e-006

9.6617e-007,1.1112e-006,1.1498e-006,1.4653e-006,1.0284e-006,2.3658e-006,3.44e-006,2.1505e-006,-3e-006,1.0465e-006,-2.3278e-005,-0.00018657,-0.00058608,-0.0013456,-0.0032989,-0.0076636,-0.011877,-0.0029001,0.039857,0.11133,0.15041,0.11133,0.039857,-0.0029001,-0.011877,-0.0076636,-0.0032989,-0.0013456,-0.00058608,-0.00018657,-2.3278e-005,1.0465e-

006,-3e-006,2.1505e-006,3.44e-006,2.3658e-006,1.0284e-006,1.4653e-006,1.1498e-006,1.1112e-006,9.6617e-007

7.9858e-007,8.6755e-007,9.0748e-007,1.1672e-006,7.3649e-007,1.283e-006,2.2805e-006,2.0997e-006,-4.2083e-006,-9.0321e-006,-2.5956e-005,-0.00013804,-0.00050528,-0.0013074,-0.0029691,-0.0063652,-0.011522,-0.012859,0.0034758,0.039857,0.061505,0.039857,0.0034758,-0.012859,-0.011522,-0.0063652,-0.0029691,-0.0013074,-0.00050528,-0.00013804,-2.5956e-005,-9.0321e-006,-4.2083e-006,2.0997e-006,2.2805e-006,1.283e-006,7.3649e-007,1.1672e-006,9.0748e-007,8.6755e-007,7.9858e-007

5.5253e-007,5.3474e-007,5.4048e-007,7.5104e-007,3.9216e-007,2.1116e-008,5.1526e-007,1.8364e-006,-3.7044e-006,-1.7534e-005,-3.172e-005,-8.2458e-005,-0.00035221,-0.0010633,-0.0022031,-0.0038527,-0.0071944,-0.01237,-0.012859,-0.0029001,0.0047415,-0.0029001,-0.012859,-0.01237,-0.0071944,-0.0038527,-0.0022031,-0.0010633,-0.00035221,-8.2458e-005,-3.172e-005,-1.7534e-005,-3.7044e-006,1.8364e-006,5.1526e-007,2.1116e-008,3.9216e-007,7.5104e-007,5.4048e-007,5.3474e-007,5.5253e-007

2.6916e-007,1.8932e-007,1.1877e-007,2.8359e-007,1.0841e-007,-8.9094e-007,-1.3925e-006,9.6001e-007,-1.025e-006,-1.8371e-005,-3.6893e-005,-4.3108e-005,-0.00017613,-0.0006975,-0.0015069,-0.0021655,-0.0034559,-0.0071944,-0.011522,-0.011877,-0.010584,-0.011877,-0.011522,-0.0071944,-0.0034559,-0.0021655,-0.0015069,-0.0006975,-0.00017613,-4.3108e-005,-3.6893e-005,-1.8371e-005,-1.025e-006,9.6001e-007,-1.3925e-006,-8.9094e-007,1.0841e-007,2.8359e-007,1.1877e-007,1.8932e-007,2.6916e-007

-9.9862e-009,-1.0504e-007,-2.6401e-007,-1.7959e-007,-9.3636e-008,-1.1515e-006,-2.7365e-006,-6.7969e-007,1.9212e-006,-1.112e-005,-3.5416e-005,-2.9411e-005,-4.5274e-005,-0.00032447,-0.00094651,-0.0015385,-0.0021655,-0.0038527,-0.0063652,-0.0076636,-0.0077233,-0.0076636,-0.0063652,-0.0038527,-0.0021655,-0.0015385,-0.00094651,-0.00032447,-4.5274e-005,-2.9411e-005,-3.5416e-005,-1.112e-005,1.9212e-006,-6.7969e-007,-2.7365e-006,-1.1515e-006,-9.3636e-008,-1.7959e-007,-2.6401e-007,-1.0504e-007,-9.9862e-009

-2.5156e-007,-3.132e-007,-5.249e-007,-5.7271e-007,-2.7825e-007,-8.4762e-007,-2.9525e-006,-2.4842e-006,2.7788e-006,-9.3668e-007,-2.4227e-005,-3.1471e-005,3.2296e-007,-6.1694e-005,-0.00040371,-0.00094651,-0.0015069,-0.0022031,-0.0029691,-0.0032989,-0.0032838,-0.0032989,-0.0029691,-0.0022031,-0.0015069,-0.00094651,-0.00040371,-6.1694e-005,3.2296e-007,-3.1471e-005,-2.4227e-005,-9.3668e-007,2.7788e-006,-2.4842e-006,-2.9525e-006,-8.4762e-007,-2.7825e-007,-5.7271e-007,-5.249e-007,-3.132e-007,-2.5156e-007

-4.3251e-007,-4.3278e-007,-6.2729e-007,-8.165e-007,-4.9656e-007,-3.6893e-007,-2.0559e-006,-3.3725e-006,1.0827e-006,5.6418e-006,-7.1228e-006,-2.8219e-005,-1.5034e-005,1.5119e-005,-6.1694e-005,-0.00032447,-0.0006975,-0.0010633,-0.0013074,-0.0013456,-0.0013095,-0.0013456,-0.0013074,-0.0010633,-0.0006975,-0.00032447,-6.1694e-005,1.5119e-005,-1.5034e-005,-2.8219e-005,-7.1228e-006,5.6418e-006,1.0827e-006,-3.3725e-006,-2.0559e-006,-3.6893e-007,-4.9656e-007,-8.165e-007,-6.2729e-007,-4.3278e-007,-4.3251e-007

-5.3861e-007,-4.844e-007,-5.9575e-007,-8.5604e-007,-7.0049e-007,-1.0627e-007,-6.981e-007,-2.7102e-006,-1.3378e-006,5.5969e-006,6.1144e-006,-1.1011e-005,-2.7044e-005,-1.5034e-005,3.2296e-007,-4.5274e-005,-0.00017613,-0.00035221,-0.00050528,-0.00058608,-0.00060511,-0.00058608,-0.00050528,-0.00035221,-0.00017613,-4.5274e-005,3.2296e-007,-1.5034e-005,-2.7044e-005,-1.1011e-005,6.1144e-006,5.5969e-006,-1.3378e-006,-2.7102e-006,-6.981e-007,-1.0627e-007,-7.0049e-007,-8.5604e-007,-5.9575e-007,-4.844e-007,-5.3861e-007

-5.6502e-007,-4.9216e-007,-4.9737e-007,-7.0907e-007,-7.7077e-007,-1.6691e-007,2.8671e-007,-9.7982e-007,-2.1799e-006,1.5526e-006,8.3023e-006,5.8937e-006,-1.1011e-005,-2.8219e-005,-3.1471e-005,-2.9411e-005,-4.3108e-005,-8.2458e-005,-0.00013804,-0.00018657,-0.00020562,-0.00018657,-0.00013804,-8.2458e-005,-4.3108e-005,-2.9411e-005,-3.1471e-005,-2.8219e-005,-

1.1011e-005,5.8937e-006,8.3023e-006,1.5526e-006,-2.1799e-006,-9.7982e-007,2.8671e-007,-1.6691e-007,-7.7077e-007,-7.0907e-007,-4.9737e-007,-4.9216e-007,-5.6502e-007

-5.1986e-007,-4.682e-007,-3.9857e-007,-4.7142e-007,-6.3672e-007,-3.4876e-007,4.944e-007,5.5881e-007,-9.5653e-007,-1.3154e-006,2.8572e-006,8.3023e-006,6.1144e-006,-7.1228e-006,-2.4227e-005,-3.5416e-005,-3.6893e-005,-3.172e-005,-2.5956e-005,-2.3278e-005,-2.2842e-005,-2.3278e-005,-2.5956e-005,-3.172e-005,-3.6893e-005,-3.5416e-005,-2.4227e-005,-7.1228e-006,6.1144e-006,8.3023e-006,2.8572e-006,-1.3154e-006,-9.5653e-007,5.5881e-007,4.944e-007,-3.4876e-007,-6.3672e-007,-4.7142e-007,-3.9857e-007,-4.682e-007,-5.1986e-007

-4.2561e-007,-4.1262e-007,-3.2748e-007,-2.6106e-007,-3.6221e-007,-3.8398e-007,1.8098e-007,9.9763e-007,7.1213e-007,-8.7173e-007,-1.3154e-006,1.5526e-006,5.5969e-006,5.6418e-006,-9.3668e-007,-1.112e-005,-1.8371e-005,-1.7534e-005,-9.0321e-006,1.0465e-006,5.4512e-006,1.0465e-006,-9.0321e-006,-1.7534e-005,-1.8371e-005,-1.112e-005,-9.3668e-007,5.6418e-006,5.5969e-006,1.5526e-006,-1.3154e-006,-8.7173e-007,7.1213e-007,9.9763e-007,1.8098e-007,-3.8398e-007,-3.6221e-007,-2.6106e-007,-3.2748e-007,-4.1262e-007,-4.2561e-007

-3.1291e-007,-3.2633e-007,-2.7058e-007,-1.4295e-007,-1.0288e-007,-2.06e-007,-1.045e-007,5.3527e-007,1.1822e-006,7.1213e-007,-9.5653e-007,-2.1799e-006,-1.3378e-006,1.0827e-006,2.7788e-006,1.9212e-006,-1.025e-006,-3.7044e-006,-4.2083e-006,-3e-006,-2.2026e-006,-3e-006,-4.2083e-006,-3.7044e-006,-1.025e-006,1.9212e-006,2.7788e-006,1.0827e-006,-1.3378e-006,-2.1799e-006,-9.5653e-007,7.1213e-007,1.1822e-006,5.3527e-007,-1.045e-007,-2.06e-007,-1.0288e-007,-1.4295e-007,-2.7058e-007,-3.2633e-007,-3.1291e-007

-2.083e-007,-2.2217e-007,-2.025e-007,-1.007e-007,2.5031e-008,2.3054e-008,-8.7992e-008,1.6457e-008,5.3527e-007,9.9763e-007,5.5881e-007,-9.7982e-007,-2.7102e-006,-3.3725e-006,-2.4842e-006,-6.7969e-007,9.6001e-007,1.8364e-006,2.0997e-006,2.1505e-006,2.1652e-006,2.1505e-006,2.0997e-006,1.8364e-006,9.6001e-007,-6.7969e-007,-2.4842e-006,-3.3725e-006,-2.7102e-006,-9.7982e-007,5.5881e-007,9.9763e-007,5.3527e-007,1.6457e-008,-8.7992e-008,2.3054e-008,2.5031e-008,-1.007e-007,-2.025e-007,-2.2217e-007,-2.083e-007

-1.2422e-007,-1.2216e-007,-1.1824e-007,-7.7148e-008,2.8742e-008,1.2239e-007,7.2448e-008,-8.7992e-008,-1.045e-007,1.8098e-007,4.944e-007,2.8671e-007,-6.981e-007,-2.0559e-006,-2.9525e-006,-2.7365e-006,-1.3925e-006,5.1526e-007,2.2805e-006,3.44e-006,3.8331e-006,3.44e-006,2.2805e-006,5.1526e-007,-1.3925e-006,-2.7365e-006,-2.9525e-006,-2.0559e-006,-6.981e-007,2.8671e-007,4.944e-007,1.8098e-007,-1.045e-007,-8.7992e-008,7.2448e-008,1.2239e-007,2.8742e-008,-7.7148e-008,-1.1824e-007,-1.2216e-007,-1.2422e-007

-5.9016e-008,-4.429e-008,-3.7546e-008,-3.7547e-008,-4.0039e-009,7.6238e-008,1.2239e-007,2.3054e-008,-2.06e-007,-3.8398e-007,-3.4876e-007,-1.6691e-007,-1.0627e-007,-3.6893e-007,-8.4762e-007,-1.1515e-006,-8.9094e-007,2.1116e-008,1.283e-006,2.3658e-006,2.7903e-006,2.3658e-006,1.283e-006,2.1116e-008,-8.9094e-007,-1.1515e-006,-8.4762e-007,-3.6893e-007,-1.0627e-007,-1.6691e-007,-3.4876e-007,-3.8398e-007,-2.06e-007,2.3054e-008,1.2239e-007,7.6238e-008,-4.0039e-009,-3.7547e-008,-3.7546e-008,-4.429e-008,-5.9016e-008

-5.8642e-009,7.4778e-009,1.7578e-008,8.6917e-009,-9.2315e-009,-4.0039e-009,2.8742e-008,2.5031e-008,-1.0288e-007,-3.6221e-007,-6.3672e-007,-7.7077e-007,-7.0049e-007,-4.9656e-007,-2.7825e-007,-9.3636e-008,1.0841e-007,3.9216e-007,7.3649e-007,1.0284e-006,1.1432e-006,1.0284e-006,7.3649e-007,3.9216e-007,1.0841e-007,-9.3636e-008,-2.7825e-007,-4.9656e-007,-7.0049e-007,-7.7077e-007,-6.3672e-007,-3.6221e-007,-1.0288e-007,2.5031e-008,2.8742e-008,-4.0039e-009,-9.2315e-009,8.6917e-009,1.7578e-008,7.4778e-009,-5.8642e-009

3.8227e-008,4.0672e-008,4.398e-008,3.7589e-008,8.6917e-009,-3.7547e-008,-7.7148e-008,-1.007e-007,-1.4295e-007,-2.6106e-007,-4.7142e-007,-7.0907e-007,-8.5604e-007,-8.165e-007,-5.7271e-007,-1.7959e-007,2.8359e-007,7.5104e-007,1.1672e-006,1.4653e-006,1.5753e-006,1.4653e-006,1.1672e-006,7.5104e-007,2.8359e-007,-1.7959e-007,-5.7271e-007,-8.165e-007,-

8.5604e-007,-7.0907e-007,-4.7142e-007,-2.6106e-007,-1.4295e-007,-1.007e-007,-7.7148e-008,-3.7547e-008,8.6917e-009,3.7589e-008,4.398e-008,4.0672e-008,3.8227e-008

7.1185e-008,6.4176e-008,5.5327e-008,4.398e-008,1.7578e-008,-3.7546e-008,-1.1824e-007,-2.025e-007,-2.7058e-007,-3.2748e-007,-3.9857e-007,-4.9737e-007,-5.9575e-007,-6.2729e-007,-5.249e-007,-2.6401e-007,1.1877e-007,5.4048e-007,9.0748e-007,1.1498e-006,1.2333e-006,1.1498e-006,9.0748e-007,5.4048e-007,1.1877e-007,-2.6401e-007,-5.249e-007,-6.2729e-007,-5.9575e-007,-4.9737e-007,-3.9857e-007,-3.2748e-007,-2.7058e-007,-2.025e-007,-1.1824e-007,-3.7546e-008,1.7578e-008,4.398e-008,5.5327e-008,6.4176e-008,7.1185e-008

9.0998e-008,8.1394e-008,6.4176e-008,4.0672e-008,7.4778e-009,-4.429e-008,-1.2216e-007,-2.2217e-007,-3.2633e-007,-4.1262e-007,-4.682e-007,-4.9216e-007,-4.844e-007,-4.3278e-007,-3.132e-007,-1.0504e-007,1.8932e-007,5.3474e-007,8.6755e-007,1.1112e-006,1.201e-006,1.1112e-006,8.6755e-007,5.3474e-007,1.8932e-007,-1.0504e-007,-3.132e-007,-4.3278e-007,-4.844e-007,-4.9216e-007,-4.682e-007,-4.1262e-007,-3.2633e-007,-2.2217e-007,-1.2216e-007,-4.429e-008,7.4778e-009,4.0672e-008,6.4176e-008,8.1394e-008,9.0998e-008

9.9373e-008,9.0998e-008,7.1185e-008,3.8227e-008,-5.8642e-009,-5.9016e-008,-1.2422e-007,-2.083e-007,-3.1291e-007,-4.2561e-007,-5.1986e-007,-5.6502e-007,-5.3861e-007,-4.3251e-007,-2.5156e-007,-9.9862e-009,2.6916e-007,5.5253e-007,7.9858e-007,9.6617e-007,1.0256e-006,9.6617e-007,7.9858e-007,5.5253e-007,2.6916e-007,-9.9862e-009,-2.5156e-007,-4.3251e-007,-5.3861e-007,-5.6502e-007,-5.1986e-007,-4.2561e-007,-3.1291e-007,-2.083e-007,-1.2422e-007,-5.9016e-008,-5.8642e-009,3.8227e-008,7.1185e-008,9.0998e-008,9.9373e-008

}

Informative references

[C1] Video Quality Experts Group, VQEG Multimedia Group Testplan version 1.16, 2007.

[C2] J. Canny, "A computational approach to edge detection", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8(6), pp. 679-698, 1986.

[C3] M. Gu and S. Eisenstat, "A Divide-and-Conquer Algorithm for the Bidiagonal SVD", in *SIAM Journal on Matrix Analysis and Applications*, Vol. 16 (issue 1), pp. 79-92, 1995.

[C4] E.R. Jessup and D.C. Sorensen, "A Parallel Algorithm for Computing the Singular Value Decomposition of a Matrix", in *SIAM Journal on Matrix Analysis and Applications*, Vol. 15 (issue 2), pp. 530-548, 1994.

[C5] C.E. McHenry, "Computation of a Best Subset in Multivariate Analysis", in *Applied Statistics*, Vol. 27 (issue 3), pp. 291-296, 1978.

[C6] C. Loeffler, A. Ligtenberg and G.S. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications", in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 988-991, 1999.

Annex D

Yonsei University full reference method

(This annex forms an integral part of this Recommendation)

D.1 Introduction

The model for objective video quality measurement is a full reference method. It is observed that the human visual system is sensitive to degradation around the edges. It is further observed that video compression algorithms tend to produce more artefacts around edge areas. Based on this observation, the model provides an objective video quality measurement method that measures degradation around the edges. In the model, an edge detection algorithm is first applied to the source video sequence to locate the edge areas. Then, the degradation of those edge areas is measured by computing the mean squared error. From this mean squared error, the edge PSNR is computed. Furthermore, the model computes two additional features which are combined with the EPSNR to produce the final video quality metric.

D.2 Full reference model based on edge degradation

Figure D.1a shows a full reference (FR) model, which takes two inputs: source video sequence (SRC) and processed video sequence (PVS). Figure D.1b shows the block-diagram of the full reference model based on edge degradation.

D.2.1 Edge PSNR (EPSNR)

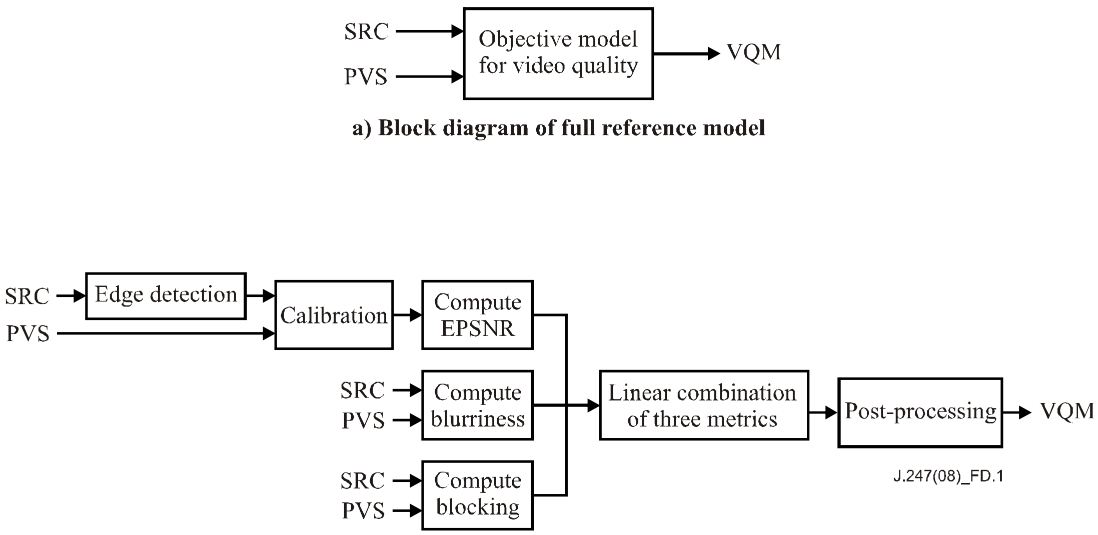
The FR models mainly measure on edge degradations. In the models, an edge detection algorithm is first applied to the source video sequence to locate the edge pixels. Then, the degradation of those edge pixels is measured by computing the mean squared error. From this mean squared error, the edge PSNR is computed.

One can use any edge detection algorithm, though there may be minor differences in the results. For example, one can use any gradient operator to locate edge pixels. A number of gradient operators have been proposed. In many edge detection algorithms, the horizontal gradient image $g_{horizontal}(m,n)$ and the vertical gradient image $g_{vertical}(m,n)$ are first computed using gradient operators. Then, the magnitude gradient image $g(m,n)$ may be computed as follows:

$$g(m,n) = |g_{horizontal}(m,n)| + |g_{vertical}(m,n)|$$

Finally, a thresholding operation is applied to the magnitude gradient image $g(m,n)$ to find edge pixels. In other words, pixels whose magnitude gradients exceed a threshold value are considered as edge pixels.

Figures D.2-D.6 illustrate the procedure. Figure D.2 shows a source image. Figure D.3 shows a horizontal gradient image $g_{horizontal}(m,n)$, which is obtained by applying a horizontal gradient operator to the source image of Figure D.2. Figure D.4 shows a vertical gradient image $g_{vertical}(m,n)$, which is obtained by applying a vertical gradient operator to the source image of Figure D.2. Figure D.5 shows the magnitude gradient image (edge image), and Figure D.6 shows the binary edge image (mask image) obtained by applying thresholding to the magnitude gradient image of Figure D.5.



a) Block diagram of full reference model

b) Block diagram of full reference model based on edge degradation

Figure D.1 – Block diagram of full reference model

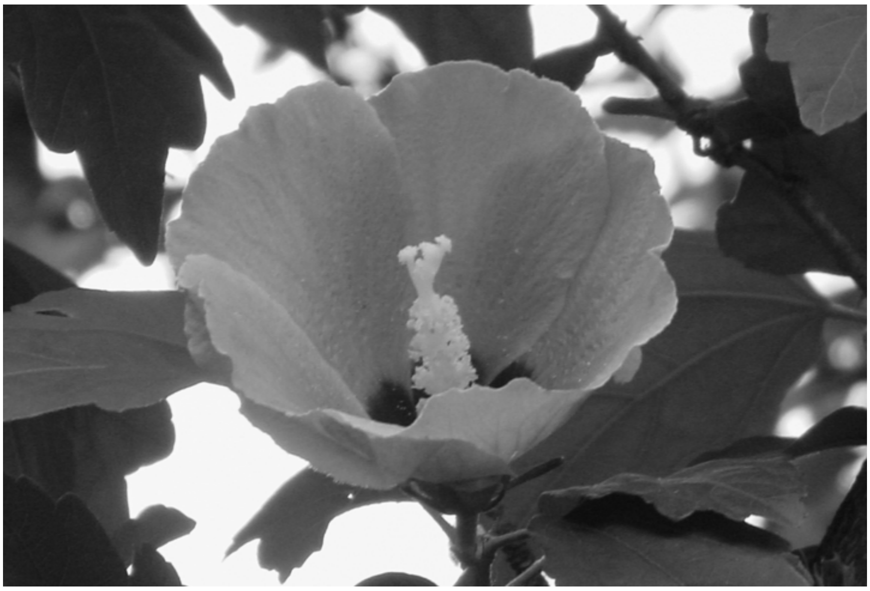


Figure D.2 – A source image (original image)

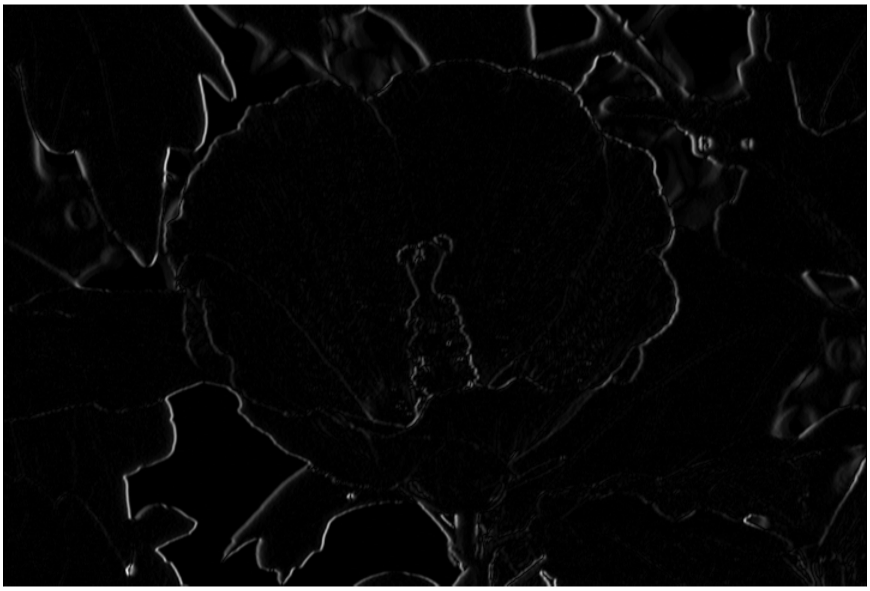


Figure D.3 – A horizontal gradient image, which is obtained by applying a horizontal gradient operator to the source image of Figure D.2

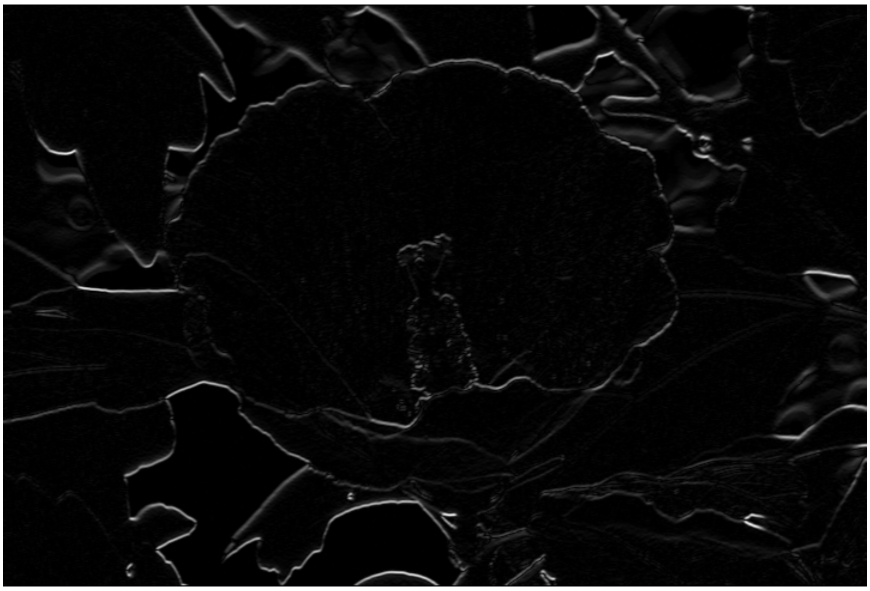


Figure D.4 – A vertical gradient image, which is obtained by applying a vertical gradient operator to the source image of Figure D.2

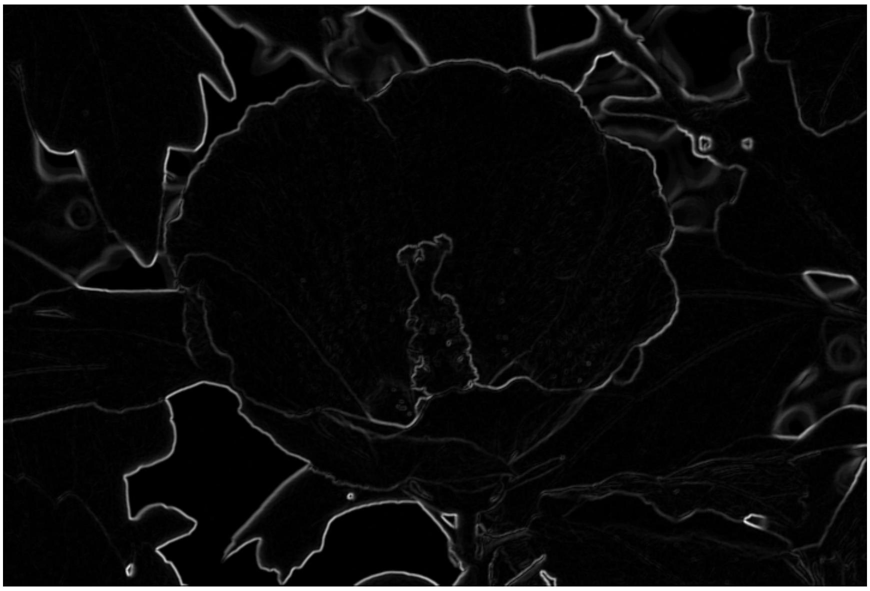


Figure D.5 – A magnitude gradient image

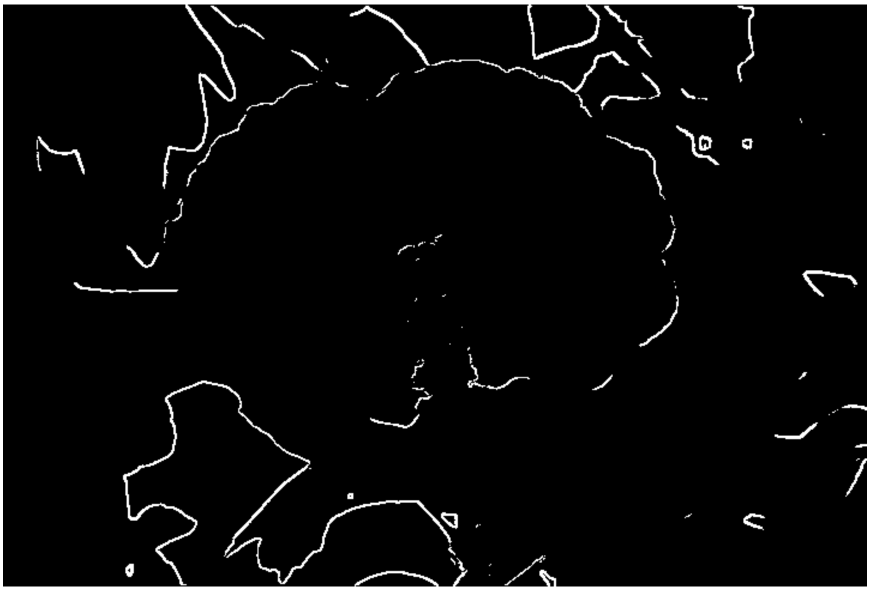


Figure D.6 – A binary edge image (mask image) obtained by applying thresholding to the magnitude gradient image of Figure D.5

Alternatively, one may use a modified procedure to find edge pixels. For instance, one may first apply a vertical gradient operator to the source image, producing a vertical gradient image. Then, a horizontal gradient operator is applied to the vertical gradient image, producing a modified successive gradient image (horizontal and vertical gradient image). Finally, a thresholding operation may be applied to the modified successive gradient image to find edge pixels. In other words, pixels of the modified successive gradient image, which exceed a threshold value, are considered as edge pixels. Figures D.7-D.9 illustrate the modified procedure. Figure D.7 shows a vertical gradient image $g_{vertical}$ (m,n), which is obtained by applying a vertical gradient operator to the source image of Figure D.2. Figure D.8 shows a modified successive gradient image (horizontal and vertical gradient image), which is obtained by applying a horizontal gradient operator to the vertical gradient image of Figure D.7. Figure D.9 shows the binary edge image (mask image) obtained by applying thresholding to the modified successive gradient image of Figure D.8.

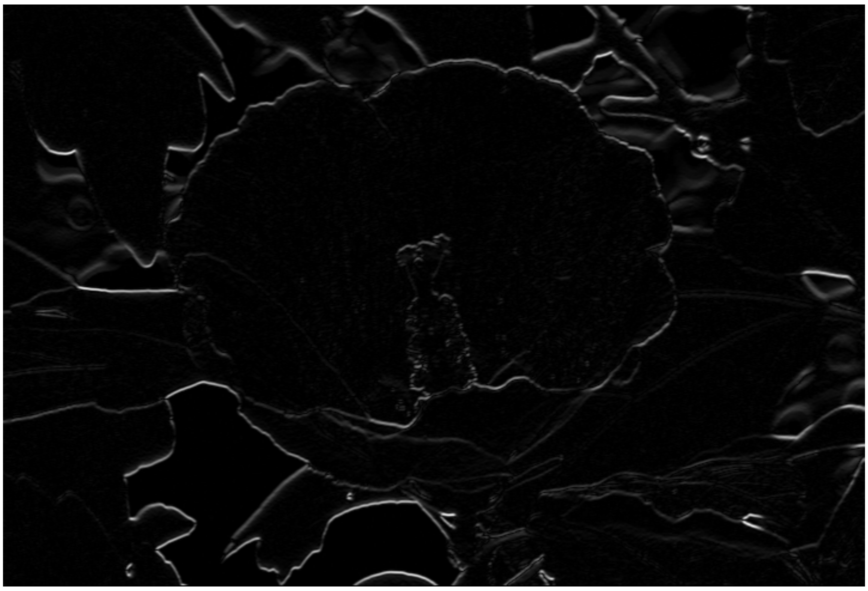


Figure D.7 – A vertical gradient image, which is obtained by applying a vertical gradient operator to the source image of Figure D.2

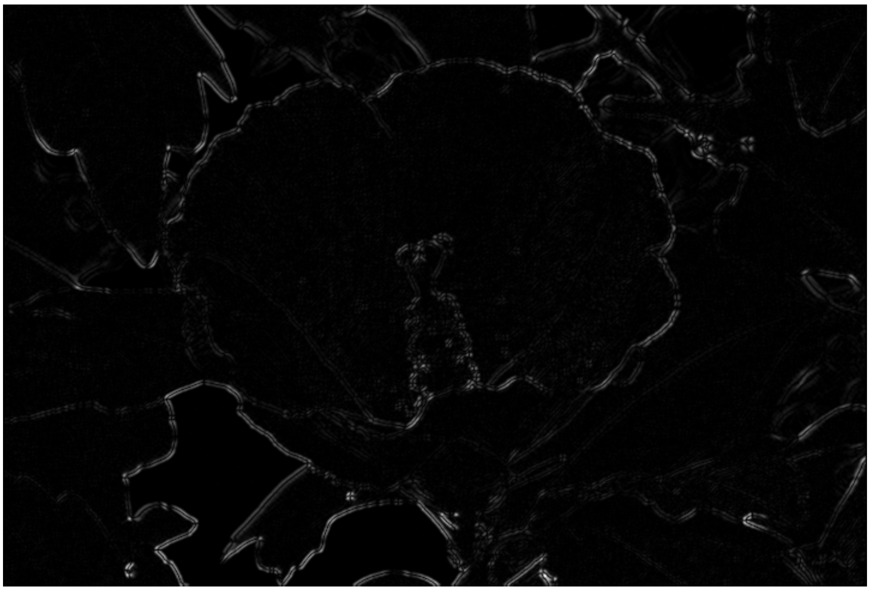


Figure D.8 – A modified successive gradient image (horizontal and vertical gradient image), which is obtained by applying a horizontal gradient operator to the vertical gradient image of Figure D.7

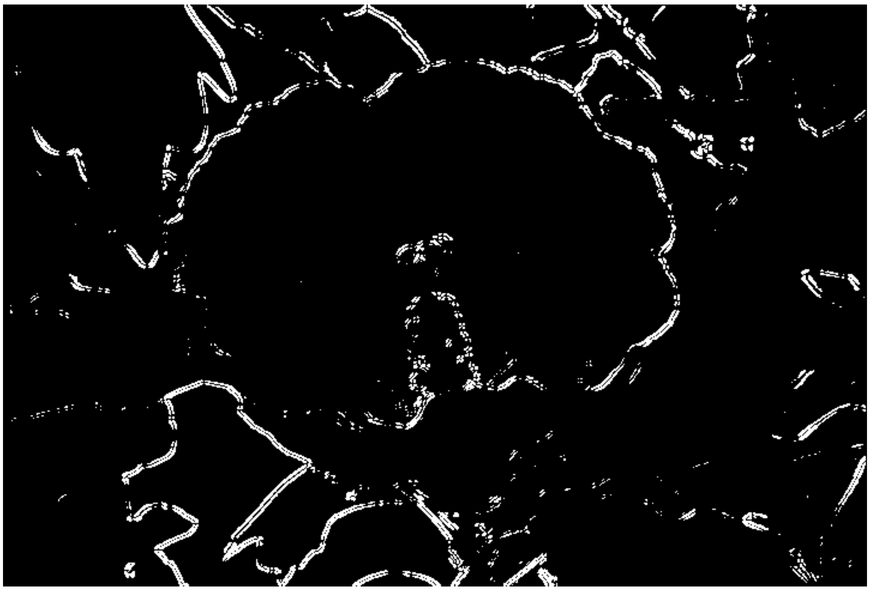


Figure D.9 – A binary edge image (mask image) obtained by applying thresholding to the modified successive gradient image of Figure D.8

It is noted that both methods can be understood as an edge detection algorithm. One may choose any edge detection algorithm depending on the nature of the video and compression algorithms. However, some methods may outperform other methods.

Thus, in the model, an edge detection operator is first applied, producing edge images (Figures D.5 and D.8). Then, a mask image (binary edge image) is produced by applying thresholding to the edge image (Figures D.6 and D.9). In other words, pixels of the edge image whose value is smaller than threshold t_e are set to zero and pixels whose value is equal to or larger than the threshold are set to a non-zero value. Figures D.6 and D.9 show some mask images. Since a video can be viewed as a sequence of frames or fields, the above-stated procedure can be applied to each frame or field of videos. Since the model can be used for field-based videos or frame-based videos, the terminology "image" will be used to indicate a field or frame.

D.2.2 Selecting edge pixels from source video sequences

In the full reference model, edge pixels are selected from each frame. However, for some video sequences, the number of edge pixels can be very small when a fixed threshold value is used. In the worst scenario, it can be zero (blank images or very low frequency images). In order to address this problem, if the number of edge pixels of an image is smaller than a given value, the user may reduce the threshold value until the number of edge pixels is larger than a given value. Alternatively, one can select edge pixels which correspond to the largest values of the horizontal and vertical gradient image. When there are no edge pixels (e.g., blank images) in a frame, one can randomly select the required number of pixels or skip the frame. For instance, if 10 edge pixels are to be selected from each frame, one can sort the pixels of the horizontal and vertical gradient image according to their values and select the largest 10 values. However, this procedure may produce multiple edge pixels at the identical locations. To address this problem, one can first select several times the desired number of pixels of the horizontal and vertical gradient image and then randomly choose the desired number of edge pixels among the selected pixels of the horizontal and vertical gradient image. In the models tested in the VQEG multimedia test, the desired number of edge pixels is randomly selected among a large pool of edge pixels. The pool of edge pixels is obtained by applying a thresholding operation to the gradient image.

It is noted that during the encoding process, cropping may be applied. In order to avoid selecting edge pixels in the cropped areas, the model selects edge pixels in the middle area (Figure D.10).

Table D.1 shows the number of edge pixels selected from each frame. Using these pixels, EPSNR is computed after spatial and temporal registrations.

Table D.1 – Number of edge pixels per frame

Video format	Size	Size after cropping	25 fps	30 fps
QCIF	176×144	168×136	111	92
CIF	352×288	338×274	264	170
VGA	640×480	614×454	379	316

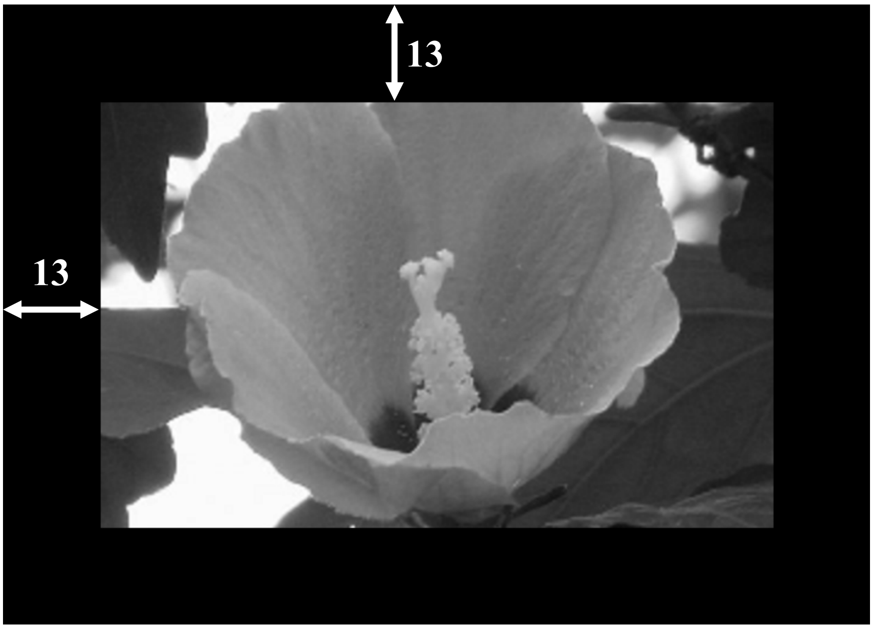


Figure D.10 – An example of cropping (VGA) and the middle area

D.2.3 Spatial/temporal registration and gain/offset adjustment

Before computing the difference between the edge pixels of the source video sequence and those of the processed video sequence, which is the received video sequence at the receiver, the model first applies a spatial/temporal registration and gain/offset adjustment. First, a full search algorithm is applied to find global spatial and temporal shifts along with gain and offset values (Figure D.11). Then, for every possible spatial shifts ($\Delta x, \Delta y$), a temporal registration using a window is performed and the EPSNR is computed. Finally, the smallest EPSNR is chosen as a video quality metric (VQM).

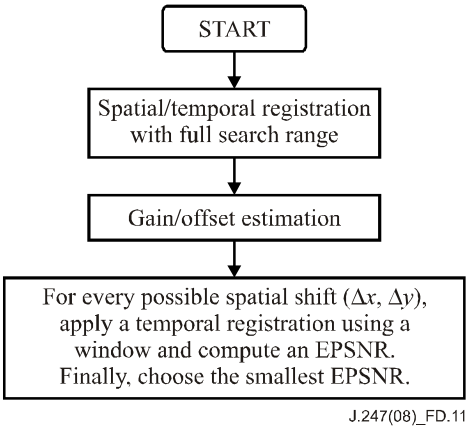

START
Spatial/temporal registration with full search range
Gain/offset estimation
For every possible spatial shift (Δx, Δy), apply a temporal registration using a window and compute an EPSNR. Finally, choose the smallest EPSNR.
J.247(08)_FD.11

Figure D.11 – Flowchart for computing EPSNR after calibration

At the monitoring point, the processed video sequence should be aligned with the edge pixels extracted from the source video sequence. The model uses a window for temporal registration. Instead of using a single frame of the processed video sequence, the model builds a window which consists of a number of adjacent frames to find the optimal temporal shift. Figure D.14 illustrates the procedure. The mean squared error within the window is computed as follows:

$$MSE_{window} = \frac{1}{N_{win}} \sum (E_{SRC}(i) - E_{PVS}(i))^2$$

where MSE_{window} is the window mean squared error, $E_{SRC}(i)$ is an edge pixel within the window which has a corresponding pixel in the processed video sequence, $E_{PVS}(i)$ is a pixel of the processed video sequence corresponding to the edge pixel, and N_{win} is the total number of edge pixels used to compute MSE_{window}. This window mean squared error is used as the difference between a frame of the processed video sequence and the corresponding frame of the source video sequence.

The window size can be determined by considering the nature of the processed video sequence (Figure D.17). For a typical application, a window corresponding to two seconds is recommended. Alternatively, various sizes of windows can be applied, and the best one which provides the smallest mean squared error can be used.

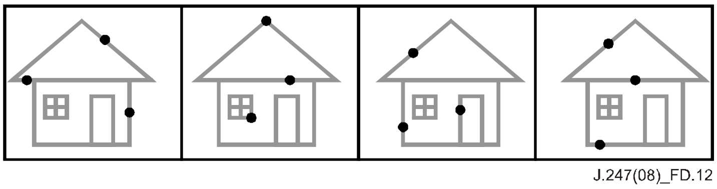


Figure D.12 – Edge pixel selection of the source video sequence

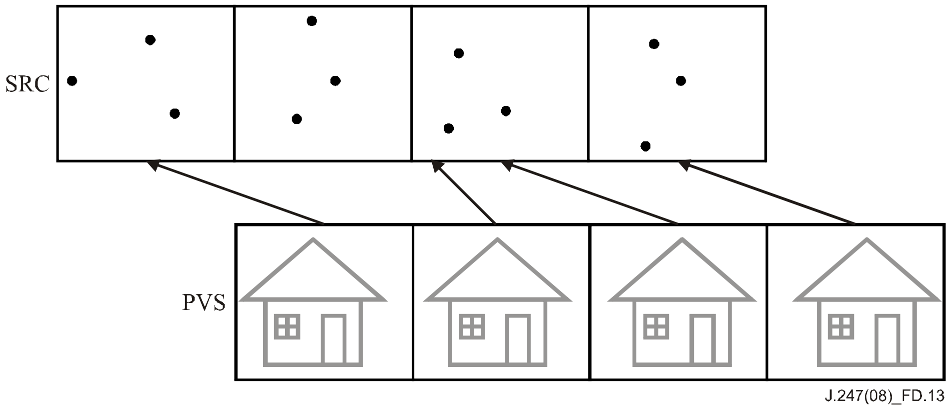


Figure D.13 – Aligning the processed video sequence to the edge pixels of the source video sequence

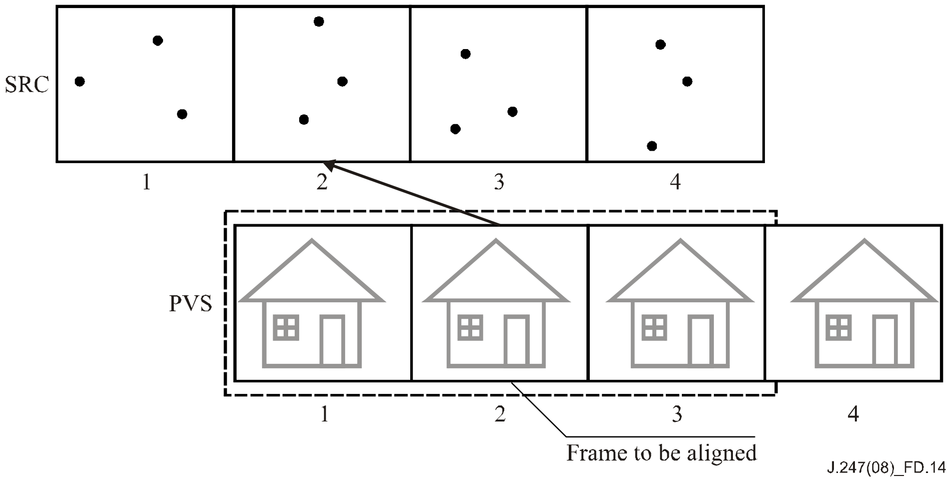
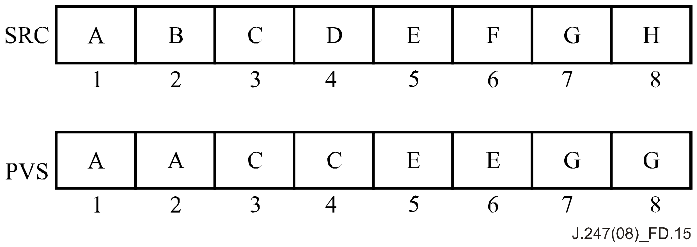


Figure D.14 – Aligning the processed video sequence to the edge pixels using a window

When the source video sequence is encoded at high compression ratios, the encoder may reduce the number of frames per second and the processed video sequence has repeated frames (Figure D.15). In Figure D.15, the processed video sequence does not have frames corresponding to some frames of the source video sequence (frames 2, 4, 6, and 8). In this case, the model does not use repeated frames in computing the mean squared error. In other words, the model performs temporal registration using the first frame (valid frame) of each block of repeated frames. Thus, in Figure D.16, only three frames (frames 3, 5, and 7) within the window are used for temporal registration.



SRC	A	B	C	D	E	F	G	H
	1	2	3	4	5	6	7	8
PVS	A	A	C	C	E	E	G	G
	1	2	3	4	5	6	7	8

Figure D.15 – Example of repeated frames

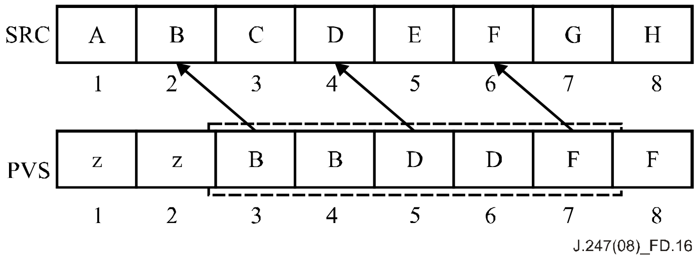


Figure D.16 – Handing repeated frames

It is possible to have a processed video sequence with irregular frame repetition, which may cause the temporal registration method using a window to produce inaccurate results. To address this problem, it is possible to locally adjust each frame of the window within a given value (e.g., ± 1) as shown in Figure D.18 after the temporal registration using a window. Then, the local adjustment which provides the minimum MSE is used to compute the EPSNR.

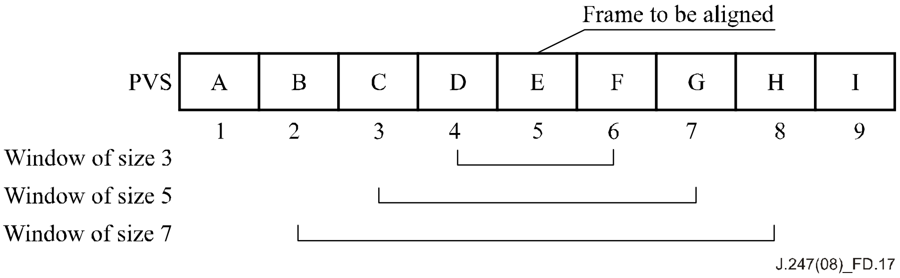


Figure D.17 – Windows of different sizes

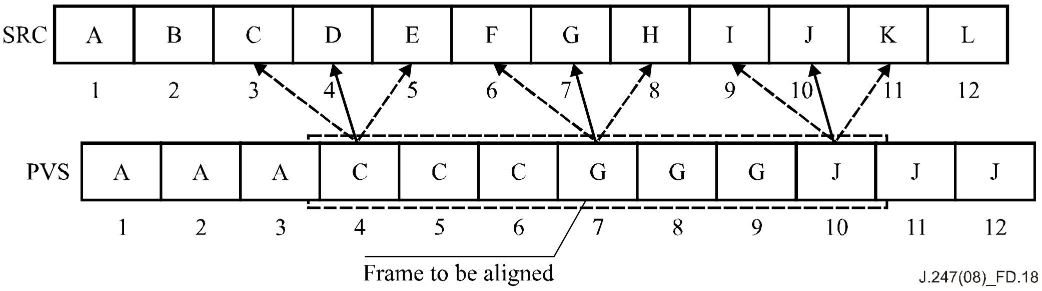


Figure D.18 – Local adjustment for temporal registration using a window

D.2.4 Computing EPSNR and post-processing

After temporal registration is performed, the average of the differences between the edge pixels of the source video sequence and the corresponding pixels of the processed video sequence is computed, which can be understood as the edge mean squared error of the processed video sequence (MSE_{edge}). Finally, the EPSNR (edge PSNR) is computed as follows:

$$EPSNR = 10\log_{10}(\frac{P^2}{MSE_{edge}})$$

where p is the peak value of the image.

In multimedia video encoding, there can be frame repeating due to reduced frame rates and frame freezing due to transmission error, which will degrade the perceptual video quality. In order to address this effect, the model applies the following adjustment before computing EPSNR:

$$MSE_{freezed\_frame\_considered} = MSE_{edge} \times \frac{K \times N_{total\_frame}}{N_{total\_frame} - N_{total\_freezed\_frame}}$$

where $MSE_{freezed\_frame\_considered}$ is the mean squared error which takes into account repeated and frozen frames, N_{total_frame} is the total number of frames, $N_{total\_freezed\_frame}$ is the total number of freezed frames, K is a constant. In the model tested in the VQEG multimedia test, K was set to 1.

When the EPSNR exceeds a certain value, the perceptual quality becomes saturated. In this case, it is possible to set the upper bound of the EPSNR. Furthermore, when a linear relationship between the EPSNR and difference mean opinion score (DMOS) is desirable, one can apply a piecewise linear function as illustrated in Figure D.19. In the model tested in the VQEG multimedia test, only the upper bound is set to 50 since polynomial curve fitting was used.

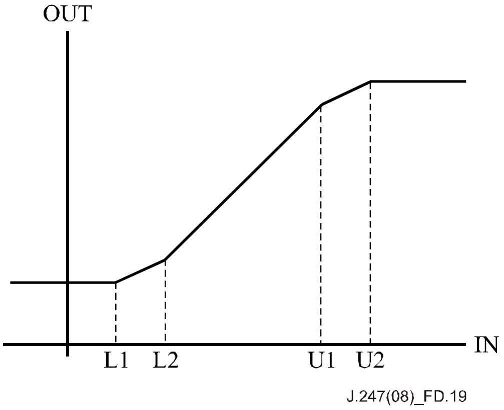


Figure D.19 – Piecewise linear function for linear relationship between the EPSNR and DMOS

D.2.5 Removing bias for linear relationship between the EPSNR and DMOS

In order to provide a linear relationship between the EPSNR and DMOS, the model estimates frames per second (FPS) by analysing the freezing frame pattern of the processed video sequence. In other words, the histogram of frozen frames is computed and the peak value is used to estimate the value of FPS. Then, based on the estimated FPS (efps), the EPSNR is adjusted as follows:

$$EPSNR_{final} = EPSNR + \alpha \quad if \quad EPSNR > \beta$$

where the values of α and β are given in Tables D.2-D.4.

Table D.2 – QCIF format

Estimated frames per second (efps)	β	α
efps<=3.0	20	–5.457
3.0<efps<6.1	28	–4.129
6.1<=efps<9.1	22	–4.401
9.1<=efps<11.3	41	–4.868
12.5<=efps<21.0	41	–3.960
21.0<=efps<2D.5	38	–3.448
2D.5<=efps<=35.0	42	–4.448

Table D.3 – CIF format

Estimated frames per second (efps)	β	α
efps<=3.0	20	–D.276
3.0<efps<6.1	29	–8.6945
6.1<=efps<9.1	20	–4.202
9.1<=efps<11.3	38	–6.550
11.3<=efps<14.5	38	–2.928
14.5<=efps<21.0	39	–3.804
21.0<=efps<2D.5	38	–4.223
2D.5<=efps<=35.0	44	–D.234

Table D.4 – VGA format

Estimated frames per second (efps)	β	α
6.0<=efps<D.5	26	–5.715
9.5<=efps<11.3	32	–5.016
14.5<=efps<21.0	36	–4.105
21.0<=efps<2D.5	38	–3.766
2D.5<=efps<=35.0	38	–3.128

D.2.6 Computing blocking and blurriness features

In order to extract features which measure the degrees of blocking and blurriness of the processed video sequence, the model first extracts edge pixels and computes the horizontal ($H(t,i,j)$) and vertical ($V(t,i,j)$) gradient component of the edge pixels. The Sobel gradient operators are recommended for this operation. From the horizontal and vertical gradient images, the magnitude ($R(t,i,j)$) and angle ($\theta(t,i,j)$) are computed as follows:

$$R(t,i,j)=\sqrt{H(t,i,j)^2+V(t,i,j)^2}$$

$$\theta(t,i,j)=\tan^{-1}\left[\frac{V(t,i,j)}{H(t,i,j)}\right]$$

Then, the horizontal and vertical component ($HV(t,i,j)$) is computed as follows:

$$HV(t,i,j)=\begin{cases} R(t,i,j), & R(t,i,j)\geq r_{\text{min}} \ \textit{and}\ m\frac{\pi}{2}-\Delta\theta<\theta(i,j,t)<m\frac{\pi}{2}+\Delta\theta \\ 0, & \textit{otherwise} \end{cases}$$

$$r_{\text{min}}=110, \quad \Delta\theta=0.225$$

From the source video sequence, the model produces a number of $HV(t,i,j)$, which is denoted as $\{HVs(k)\}$. It is noted that all pixels which satisfy the condition ($r_{\text{min}}\geq 110$) are used in this procedure. After the spatial and temporal registration described previously, the model finds the PVS edge pixels corresponding to the SRC edge pixels. By applying the procedure to the PVS edge pixels, the model generates a number of $HV(t,i,j)$, which is denoted as $\{HVp(k)\}$.

Finally, the blocking feature ($F_{blocking}$) is computed as follows:

$$F_{blocking} = \frac{1}{n_{blocking}} \sum_k \left(HVp(k) - HVs(k)\right) \ \ if \ \ \left(HVp(k) > HVs(k)\right)$$

where $n_{blocking}$ is the number of pixels satisfying the condition $\left(HVp(k) > HVs(k)\right)$. It may be desirable to set upper and lower bounds on the blocking feature. For a block of repeated frames, the difference of the first frame of the block of repeated frames is used for the remaining frames of the block of the repeated frames.

Furthermore, the blurriness feature (F_{blur}) is computed as follows:

$$F_{blur} = \frac{1}{n_{blur}} \sum_k \left(HVs(k) - HVp(k)\right) \ \ if \ \ \left(HVs(k) > HVp(k)\right)$$

where n_{blur} is the number of pixels satisfying the condition $\left(HVs(k) > HVp(k)\right)$. For a block of repeated frames, the difference of the first frame of the block of repeated frames is used for the remaining frames of the block of the repeated frames. It may be desirable to set upper and lower bounds on the blurriness feature.

D.2.7 Computation of final VQM

The VQM is computed by a linear combination of the three features (EPSNR, $F_{blocking}$, F_{blur}) as follows:

$$VQM = EPSNR_{final} + w_1 \times F_{blocking} + w_2 \times F_{blur}$$

In the VQEG test, the following values are used:

$$w_1 = -1/14$$

$$w_2 = -1/14$$

D.3 Conclusions

A new model for the objective measurement of the video quality is proposed based on edge degradation, blurring and blocking features. Therefore, the model is suited to applications which require video quality evaluation and quality monitoring.

Appendix I

Excerpts from the synopsis from the Video Quality Experts Group on the validation of objective models of multimedia quality assessment, phase I

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

I.1 Introduction

This appendix presents results from the Video Quality Experts Group (VQEG) multimedia validation testing of objective video quality models for mobile/PDA and broadband Internet communication services.[11]

The multimedia (MM) test contains two parallel evaluations of test video material. One evaluation is by panels of human observers (i.e., subjective testing). The other is by objective computational models of video quality (i.e., proponent models). The objective models are meant to predict the subjective judgments. Each subjective test is referred to as an "experiment" throughout this appendix.

The MM test discussed addresses three video resolutions (VGA, CIF and QCIF) and three types of models: full reference (FR), reduced reference (RR) and no reference (NR). FR models have full access to the source video; RR models have limited bandwidth access to the source video; and NR models do not have access to the source video. RR models can be used in certain applications that cannot be addressed by FR models, such as in-service monitoring in networks. NR models can be used in certain applications that cannot be addressed by FR or RR approaches. Typically, no-reference models are applied in situations where the user does not have access to the source. Proponents were given the option of submitting different models for each video resolution and model type.

Forty one subjective experiments provided data against which model validation was performed. The experiments were divided among the three video resolutions and two frame rates (25 fps and 30 fps). A common set of carefully chosen video sequences were inserted identically into each experiment at a given resolution, to anchor the video experiments to one another and assist in comparisons between the subjective experiments. The subjective experiments included processed video sequences with a wide range of quality, and both compression and transmission errors were present in the test conditions. These 41 subjective experiments included 346 source video sequences and 5320 processed video sequences. These video clips were evaluated by 984 viewers.

A total of thirteen organizations performed subjective MM testing. Of these organizations, five were model proponents (NTT, OPTICOM, Psytechnics, SwissQual and Yonsei University) and the remainder were either independent testing laboratories (Acreo, CRC, IRCCyN, France Telecom, FUB, Nortel, NTIA and Verizon) or laboratories that helped by running processed video sequences (PVS) and subjective experiments (KDDI and Symmetricom). Objective models were submitted prior to scene selection, PVS generation and subjective testing, to ensure none of the models could be trained on the test material. Of the 31 models submitted, six were withdrawn; therefore, 25 are presented in this appendix. A model is considered in this context to be a model type (i.e., FR or RR or NR) for a specified resolution (i.e., VGA or CIF or QCIF).

[11] This appendix has been adapted from [VQEG] with permission. This synopsis is a shortened version of the VQEG MM synopsis, as this appendix only addresses FR models.

Results for models submitted by the following proponent organizations are included in this Appendix:

- NTT (Japan)
- OPTICOM (Germany)
- Psytechnics (UK)
- Yonsei University (Korea)

VQEG cautions that the MM data should not be used as evidence to standardize any other objective video quality model that was not tested within this phase. Such a comparison would not be valid, because another model could have been trained on the MM data.

I.2 Model performance evaluation techniques

The models were evaluated using three statistics that provide insights into model performance: Pearson correlation, root-mean squared error (RMSE) and outlier ratios. These statistics compare the objective model's predictions with the subjective quality as judged by a panel of human observers. Each model was fitted to each subjective experiment, by optimizing Pearson correlation with subjective data first, and minimizing RMSE second.

Each of these statistics (Pearson correlation, RMSE and outlier ratios) can be used to determine whether a model is in the group of top performing models for one video format/resolution (i.e., a group of models that include the top performing model and models that are statistically equivalent to the top performing model). Note that a model that is not in the top performing group and is statistically worse than the top performing model but may be statistically equivalent to one or more of the models that are in the top performing group. Statistical significances are computed for each metric separately, and therefore the models' ranking per video resolution is accomplished per each statistical metric.

When examining the total number of times a model is statistically equivalent to the top performing model for each resolution, comparisons between models should be performed carefully. Determining which differences in totals are statistically significant requires additional analysis not available in this appendix. As a general guideline, small differences in these totals do not indicate an overall difference in performance.

Primary analysis considers each video sequence separately. Secondary analysis averages over all video sequences associated with each video system (or condition), and thus reflects how well the model tracks the average hypothetical reference circuit (HRC) performance. The common set of video sequences are included in primary analysis but eliminated from secondary analysis. The following clauses report on model performance across model type and resolution. The reader should be aware that performance is reported according to primary evaluation metrics and secondary evaluation metrics. Secondary analysis is presented to supplement the primary analysis. The primary analysis is the most important determinant of a model's performance.

PSNR was computed as a reference measure, and compared to all models. PSNR was computed using an exhaustive search for calibration and one constant delay for each video sequence. Models were required to perform their own calibration, where needed. While PSNR serves as a references measure, it is not necessarily the most useful benchmark for recommendation of models.

I.3 FR model performance

FR model results from NTT, OPTICOM, Psytechnics and Yonsei for all three resolutions (VGA, CIF and QCIF) are included in this report.

I.3.1 Primary analysis of FR models

The average correlations of the primary analysis for the FR VGA models ranged from 0.79 to 0.83, and PSNR was 0.71. Individual model correlations for some experiments were as high as 0.94. The average RMSE for the FR VGA models ranged from 0.57 to 0.62, and PSNR was 0.71. The average outlier ratio for the FR VGA models ranged from 0.50 to 0.54, and PSNR was 0.62. All proposed models performed statistically better than PSNR for at least 8 of the 13 experiments. Based on each metric, each FR VGA model was in the group of top performing models the following number of times:

Statistic	Psy_FR	Opt_FR	Yon_FR	NTT_FR	PSNR
Correlation	11	10	10	8	3
RMSE	10	8	6	4	0
Outlier ratio	12	11	8	9	4

The average correlations of the primary analysis for the FR CIF models ranged from 0.78 to 0.84, and PSNR was 0.66. Individual model correlations for some experiments were as high as 0.92. The average RMSE for the FR CIF models ranged from 0.53 to 0.60, and PSNR was 0.72. The average outlier ratio for the FR CIF models ranged from 0.51 to 0.54, and PSNR was 0.63. All proposed models performed statistically better than PSNR for at least 10 of the 14 experiments. Based on each metric, each FR CIF model was in the group of top performing models the following number of times:

Statistic	Psy_FR	Opt_FR	Yon_FR	NTT_FR	PSNR
Correlation	14	13	10	8	0
RMSE	13	10	9	6	0
Outlier ratio	12	13	11	10	1

The average correlations of the primary analysis for the FR QCIF models ranged from 0.76 to 0.84, and PSNR was 0.66. Individual model correlations for some experiments were as high as 0.94. The average RMSE for the FR QCIF models ranged from 0.52 to 0.62, and PSNR was 0.72. The average outlier ratio for the FR QCIF models ranged from 0.46 to 0.52, and PSNR was 0.60. All proposed models performed statistically better than PSNR for at least 8 of the 14 experiments. Based on each metric, each FR QCIF model was in the group of top performing models the following number of times:

Statistic	Psy_FR	Opt_FR	Yon_FR	NTT_FR	PSNR
Correlation	12	11	4	9	1
RMSE	11	10	2	7	1
Outlier ratio	12	11	8	10	4

The gaps in performance between all of the models for individual experiments are very small. The models from Psytechnics and OPTICOM are observed to perform slightly better than the NTT and Yonsei models in some resolutions; however, for some experiments, this difference is not statistically significant. The Psytechnics and OPTICOM models usually produce statistically equivalent results. For QCIF, the model from NTT is often statistically equivalent to the models of

Psytechnics and OPTICOM. For VGA, the Yonsei model is typically statistically equivalent to the Psytechnics and OPTICOM models.

I.3.2 Secondary analysis of FR models

The secondary analysis shows in principle a similar picture. The correlation coefficients generally increase. For VGA, the FR models from OPTICOM and Psytechnics are observed to perform slightly better than the two other ones. However, all tested models show disadvantages for individual experiments. For CIF, the performance of all FR models is very similar. For QCIF, the performance of all FR models is very similar. The NTT model shows no disadvantages for any experiment (all correlation coefficients above 0.90).

I.3.3 FR model conclusions of VQEG

– Some of the FR models may be considered for standardization, making sure that the scopes of these Recommendations are written carefully to ensure that the use of the models is defined appropriately.

– If the scope of these Recommendations includes video system comparisons (e.g., comparing two codecs), then the Recommendation should include instructions indicating how to perform an accurate comparison.

– None of the evaluated models reached the accuracy of the normative subjective testing.

– All of the FR models performed statistically better than PSNR.

– The secondary analysis requires averaging over a well defined set of sequences while the tested system, including all processing steps for the video sequences, must remain exactly the same for all clips. Averaging over arbitrary sequences will lead to much worse results.

It should be noted that in case of new coding and transmission technologies, which were not included in this evaluation; the objective models can produce erroneous results. Here a subjective evaluation is required.

I.4 Data analysis executed by ILG

Subjective data included virtually in this appendix is being made available by the Video Quality Experts Group (VQEG) to assist the research community. Statistics from the VQEG synopsis can be used in papers by anyone provided that identification that the VQEG synopsis was the source of the data is made explicitly in such papers.

VQEG validation subjective experiment data is placed in the public domain; however, the video sequences themselves are only available for further experiments from the content provider and with restrictions required by the relevant copyright holder for the particular video sequence. VQEG objective validation test data may only be used with the proponent's approval. Interested parties should contact the VQEG for additional information. Nevertheless, any summary data contained in this appendix is available for users of this Recommendation.

The ILG has analysed the data collected by the VQEG and provided it to ITU for distribution with this Recommendation. The official ILG data analysis is provided in the associated file indicated below.

The associated zip file for this Recommendation contains the following files in the Software folder:

General instructions:	readme.txt
Analysis of data:	performance_analysis.xls
Copyright notice:	copyright_notice.txt

That zip file is available for free download here:

www.itu.int/rec/T-REC-J.247

Appendix II

Equations for model evaluation metrics

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

Evaluation metrics

Pearson correlation coefficient

The Pearson correlation coefficient *R* (see equation II.1) measures the linear relationship between a model's performance and the subjective data. Its great virtue is that it is on a standard, comprehensible scale of –1 to 1 and that it has been used frequently in similar testings.

$$R = \frac{\sum_{i=1}^{N}(Xi-\overline{X})*(Yi-\overline{Y})}{\sqrt{\sum(Xi-\overline{X})^2}*\sqrt{\sum(Yi-\overline{Y})^2}} \qquad \text{(II.1)}$$

Xi denotes the subjective score (*DMOS*(*i*) for FR models) and *Yi* the objective score ($DMOS_p(i)$ for FR). N in equation II.1 represents the total number of video clips considered in the analysis.

Therefore, in the context of this test, the value of *N* in equation II.1 is:

- *N*=152 for FR (=166–14 since the evaluation for FR/RR discards the reference videos and there are 14 reference videos in each experiment).
- Note that if any PVS in the experiment was discarded for data analysis, then the value of *N* changes accordingly.

The sampling distribution of Pearson's *R* is not normally distributed. "Fisher's *z* transformation" converts Pearson's *R* to the normally distributed variable *z*. This transformation is given by the following equation:

$$z = 0.5 \cdot \ln\left(\frac{1+R}{1-R}\right)$$

The statistic of *z* is approximately normally distributed and its standard deviation is defined by:

$$\sigma_z = \sqrt{\frac{1}{N-3}} \qquad \text{(II.2)}$$

The 95% confidence interval (*CI*) for the correlation coefficient is determined using the Gaussian distribution, which characterizes the variable *z* and it is given by:

$$\mathrm{CI} = \pm \mathrm{K1}*\sigma_z \qquad \text{(II.3)}$$

NOTE 1 – For a Gaussian distribution, *K*1 = 1.96 for the 95% confidence interval. If *N*<30 samples are used then the Gaussian distribution must be replaced by the appropriate Student's *t* distribution, depending on the specific number of samples used.

Therefore, in the context of this test, *K*1 = 1.96.

The lower and upper bound associated to the 95% confidence interval (*CI*) for the correlation coefficient is computed for the Fisher's *z* value:

$$LowerBound = z - K1*\sigma_z$$

$$UpperBound = z + K1*\sigma_z$$

NOTE 2 – The values of Fisher's *z* of lower and upper bounds are then converted back to Pearson's *R* to get the *CI* of correlation *R*.

Root mean square error

The accuracy of the objective metric is evaluated using the root mean square error (rmse) evaluation metric.

The difference between measured and predicted DMOS is defined as the absolute prediction error *Perror*:

$$Perror(i) = DMOS(i) - DMOS_p(i) \tag{II.4}$$

where the index *i* denotes the video sample.

The root mean square error of the absolute prediction error *Perror* is calculated with the equation:

$$rmse = \sqrt{\left(\frac{1}{N-d}\sum_N Perror[i]^2\right)} \tag{II.5}$$

where *N* denotes the total number of video clips considered in the analysis and *d* the number of degrees of freedom of the mapping function.

In the case of a data fitting using a third-order monotonic polynomial function, *d*=4 (since there are 4 coefficients in the fitting function).

In the context of this test plan, the value of *N* in equation II.5 is:

- *N*=152 for FR models (since the evaluation discards the reference videos and there are 14 reference videos in each experiment).
- Note that if any PVS in the experiment is discarded for data analysis, then the value of *N* changes accordingly.

The root mean square error is approximately characterized by a χ^2 (*n*) [Spiegel], where *n* represents the degrees of freedom and it is defined by:

$$n = N - d \tag{II.6}$$

where *N* represents the total number of samples.

Using the $\chi^{\wedge 2}$ (n) distribution, the 95% confidence interval for the rmse is given by equation II.7 [Spiegel]:

$$\frac{rmse * \sqrt{N-d}}{\sqrt{\chi^2_{0.025}(N-d)}} < rmse < \frac{rmse * \sqrt{N-d}}{\sqrt{\chi^2_{0.975}(N-d)}} \tag{II.7}$$

Outlier ratio (using standard error of the mean)

The consistency attribute of the objective metric is evaluated by the outlier ratio (OR) which represents number of "outlier-points" to total points *N*:

$$OR = \frac{TotalNoOutliers}{N} \tag{II.8}$$

where an outlier is a point for which

$$|Perror(i)| > K2 * \frac{\sigma(DMOS(i))}{\sqrt{Nsubjs}} \tag{II.9}$$

where σ(*DMOS*(*i*)) represents the standard deviation of the individual scores associated with the video clip *i*, and *Nsubjs* is the number of viewers per video clip *i*. In this test plan, a number of 24 viewers (*Nsubjs*=24) per video clip was used.

NOTE 3 – *DMOS*(*i*) is used for FR models.

NOTE 4 – For a Gaussian distribution, *K*2 = 1.96 for the 95% confidence interval. If the mean (DMOS) is based on less than thirty samples (i.e., *Nsubjs* < 30), then the Gaussian distribution must be replaced by the appropriate Student's *t* distribution, depending on the specific number of samples in the mean [Spiegel]. In the case of 24 viewers per video (i.e., the number of samples in the mean is 24), the number of degrees of freedom is *df*=23 and therefore the associated *K*2 = 2.069 is used for the 95% confidence interval.

Therefore, in the context of this test plan, *K*2 = 2.069.

The outlier ratio represents the proportion of outliers in *N* number of samples. Thus, the binomial distribution could be used to characterize the outlier ratio. The outlier ratio is represented by a distribution of proportions [Spiegel] characterized by the mean *p* (equation II.10) and standard deviation σ_p (equation II.11).

$$p = \frac{TotalNoOutliers}{N} \tag{II.10}$$

$$\sigma_p = \sqrt{\frac{p^*(1-p)}{N}} \tag{II.11}$$

where *N* is the total number of video clips considered in the analysis.

For *N*>30, the binomial distribution, which characterizes the proportion *p*, can be approximated with the Gaussian distribution. Therefore, the 95% confidence interval (*CI*) of the outlier ratio is given by:

$$CI = \pm\ 1.96^*\sigma_{\mathrm{p}} \tag{II.12}$$

NOTE 5 – If the mean is based on less than thirty samples (i.e., *N* < 30), then the Gaussian distribution must be replaced by the appropriate Student's *t* distribution, depending on the specific number of samples in the mean [Spiegel].

Informative references

[Spiegel] M. Spiegel, "Schaum's Outline of Theory and Problems of Statistics", McGraw Hill, 1998.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	**Cable networks and transmission of television, sound programme and other multimedia signals**
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	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	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2010