

RECOMMENDATION ITU-R BT.1683

Objective perceptual video quality measurement techniques for standard definition digital broadcast television in the presence of a full reference

(Question ITU-R 44/6)

(2004)

The ITU Radiocommunication Assembly,

considering

- a) that the ability to measure automatically the quality of broadcast video has long been recognized as a valuable asset to the industry;
- b) that conventional objective methods are no longer fully adequate for measuring the perceived video quality of digital video systems using compression;
- c) that objective measurement of perceived video quality will complement conventional objective test methods;
- d) that current formal subjective assessment methods are time-consuming and expensive and generally not suited for operational conditions;
- e) that objective measurement of perceived video quality may usefully complement subjective assessment methods,

recommends

- 1 that the guidelines, scope and limitations given in Annex 1 be used in the application of the objective video quality models found in Annexes 2-5;
- 2 that the objective video quality models given in Annexes 2-5 be used for objective measurement of perceived video quality.

Annex 1**Summary**

This Recommendation specifies methods for estimating the perceived video quality of a one-way video transmission system. This Recommendation applies to baseband signals. The estimation methods in this Recommendation are applicable to:

- codec evaluation, specification, and acceptance testing;
- potentially real-time, in-service quality monitoring at the source;
- remote destination quality monitoring when a copy of the source is available;
- quality measurement of a storage or transmission system that utilizes video compression and decompression techniques, either a single pass or a concatenation of such techniques.

Introduction

The ability to measure automatically the quality of broadcast video has long been recognized as a valuable asset to the industry. The broadcast industry requires such tools to replace or supplement costly and time-consuming subjective quality testing. Traditionally, objective quality measurement has been obtained by calculating peak signal-to-noise ratios (PSNRs). Although a useful indicator

of quality, PSNR has been shown to be a less than satisfactory representation of perceptual quality. To overcome the limitations associated with PSNR, research has been directed towards defining algorithms that can measure the perceptual quality of broadcast video. Such objective perceptual quality measurement tools may be applied to testing the performance of a broadcast network, as equipment procurement aids and in the development of new broadcast video coding techniques. In recent years, significant work has been dedicated to the development of reliable and accurate tools that can be used to objectively measure the perceptual quality of broadcast video. This Recommendation defines objective computational models that have been shown to be superior to PSNR as automatic measurement tools for assessing the quality of broadcast video. The models were tested on 525-line and 625-line material conforming to Recommendation ITU-R BT.601, which was characteristic of secondary distribution of digitally encoded television quality video.

The performance of the perceptual quality models was assessed through two parallel evaluations of the test video material¹. In the first evaluation, a standard subjective method, the double stimulus continuous quality scale (DSCQS) method, was used to obtain subjective ratings of quality of video material by panels of human observers (Recommendation ITU-R BT.500 – Methodology for the subjective assessment of the quality of television pictures). In the second evaluation, objective ratings were obtained by the objective computational models. For each model, several metrics were computed to measure the accuracy and consistency with which the objective ratings predicted the subjective ratings. Three independent laboratories conducted the subjective evaluation portion of the test. Two laboratories, Communications Research Center (CRC, Canada) and Verizon (United States of America), performed the test with 525/60 Hz sequences and a third lab, Fondazione Ugo Bordoni (FUB, Italy), performed the test with 625/50 Hz sequences. Several laboratories “proponents” produced objective computational models of the video quality of the same video sequences tested with human observers by CRC, Verizon and FUB. The results of the tests are given in Appendix 1.

This Recommendation includes the objective computational models shown in Table 1.

TABLE 1

Model number	Name	Video Quality Experts Group (VQEG) proponent	Country	Annex
1	British Telecom	D	United Kingdom	2
2	Yonsei University/Radio Research Laboratory/SK Telecom	E	Korea (Rep. of)	3
3	Center for Telecommunications Research and Development (CPqD)	F	Brazil	4
4	National Telecommunications and Information Administration/Institute for Telecommunication Science (NTIA/ITS)	H	United States of America	5

¹ ITU-R Doc. 6Q/14 [September, 2003] Final Report from the Video Quality Experts Group on the Validation of Objective Models of Video Quality Assessment, Phase II (FR-TV2).

A complete description of the above four objective computational models is provided in Annexes 2-5.

Existing video quality test equipment can be used until new test equipment implementing any of the four above models is readily available.

For any model to be considered for inclusion in the normative section of this Recommendation in the future, the model must be verified by an open independent body (such as VQEG) which will do the technical evaluation within the guidelines and performance criteria set out by Radiocommunication Study Group 6. The intention of Radiocommunication Study Group 6 is to eventually recommend only one normative full reference method.

1 Scope

This Recommendation specifies methods for estimating the perceived video quality of a one-way video system. This Recommendation applies to baseband signals. The objective video performance estimators are defined for the end-to-end quality between the two points. The estimation methods are based on processing 8-bit digital component video as defined by Recommendation ITU-R BT.601². The encoder can utilize various compression methods (e.g. Moving Picture Experts Group (MPEG), ITU-T Recommendation H.263, etc.). The models proposed in this Recommendation may be used to evaluate a codec (encoder/decoder combination) or a concatenation of various compression methods and memory storage devices. While the derivation of the objective quality estimators described in this Recommendation might have considered error impairments (e.g. bit errors, dropped packets), independent testing results are not currently available to validate the use of the estimators for systems with error impairments. The validation test material did not contain channel errors.

1.1 Application

This Recommendation provides video quality estimations for television video classes (TV0-TV3), and multimedia video class (MM4) as defined in ITU-T Recommendation P.911, Annex B. The applications for the estimation models described in this Recommendation include but are not limited to:

- codec evaluation, specification, and acceptance testing, consistent with the limited accuracy as described below;
- potentially real-time, in-service quality monitoring at the source;
- remote destination quality monitoring when a copy of the source is available;
- quality measurement of a storage or transmission system that utilizes video compression and decompression techniques, either a single pass or a concatenation of such techniques.

1.2 Limitations

The estimation models described in this Recommendation cannot be used to replace subjective testing. Correlation values between two carefully designed and executed subjective tests (i.e. in two different laboratories) normally fall within the range 0.92 to 0.97. This Recommendation does not

² This does not preclude implementation of the measurement method for one-way video systems that utilize composite video input and outputs. Specification of the conversion between composite and component domains is not part of this Recommendation. For example, SMPTE 170M Standard specifies one method for performing this conversion for NTSC.

supply a means for quantifying potential estimation errors. Users of this Recommendation should review the comparison of available subjective and objective results to gain an understanding of the range of video quality rating estimation errors.

The predicted performance of the estimation models is not currently validated for video systems with transmission channel error impairments.

Annex 2

Model 1

CONTENTS

	<i>Page</i>
1 Introduction	5
2 BTFR	5
3 Detectors	5
3.1 Input conversion	6
3.2 Crop and offset	6
3.3 Matching	7
3.3.1 Matching statistics.....	9
3.3.2 MPSNR	9
3.3.3 Matching Vectors	9
3.4 Spatial frequency analysis	10
3.4.1 Pyramid transform.....	10
3.4.2 Pyramid SNR	12
3.5 Texture analysis	12
3.6 Edge analysis	13
3.6.1 Edge detection.....	13
3.6.2 Edge differencing.....	13
3.7 MPSNR analysis	14
4 Integration.....	14
5 Registration.....	15
6 References	15
Annex 2a	16

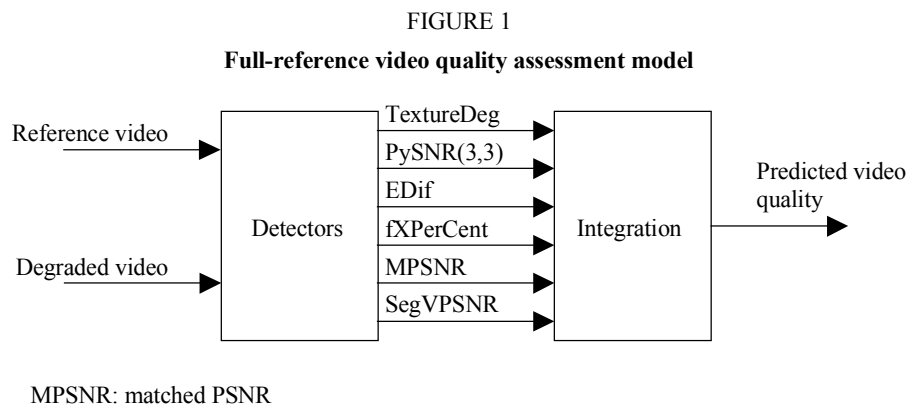
1 Introduction

The BT full-reference (BTFR) automatic video quality assessment tool produces predictions of video quality that are representative of human quality judgements. This objective measurement tool digitally simulates features of the human visual system (HVS) to give accurate predictions of video quality and offers a viable alternative to costly and time-consuming formal subjective assessments.

A software implementation of the model was entered in the VQEG2 tests and the resulting performance presented in a test report.

2 BTFR

The BTFR algorithm consists of detection followed by integration as shown in Fig. 1. Detection involves the calculation of a set of perceptually meaningful detector parameters from the undistorted (reference) and distorted (degraded) video sequences. These parameters are then input to the integrator, which produces an estimate of the perceived video quality by appropriate weighting. The choice of detectors and weighting factors are founded on knowledge of the spatial and temporal masking properties of the HVS and determined through calibration experiments.



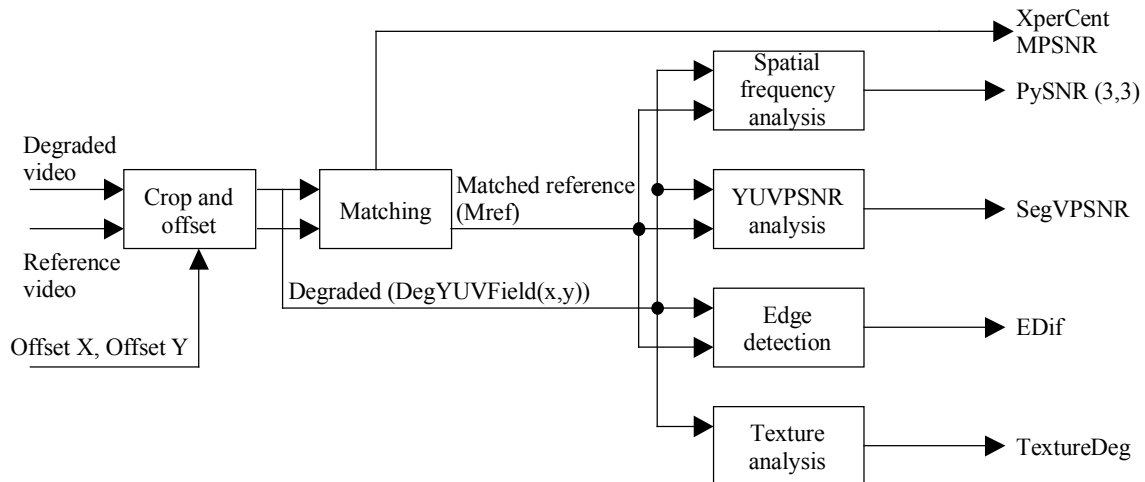
1683-01

Input video of types 625 (720×576) interlaced at 50 fields/s and 525 (720×486) interlaced at 59.94 fields/s in *YUV422* format are supported by the model.

3 Detectors

The detection module of the BTFR algorithm calculates a number spatial, temporal and frequency-based measures from the input *YUV* formatted sequences, as shown in Fig. 2.

FIGURE 2
Detection



1683-02

3.1 Input conversion

First, the input sequences are converted from *YUV422* interlaced format to a block *YUV444* deinterlaced format so that each successive field is represented by arrays *RefY*, *RefU* and *RefV*:

$$RefY(x, y) \quad x = 0 \dots X - 1, \quad y = 0 \dots Y - 1 \quad (1)$$

$$RefU(x, y) \quad x = 0 \dots X - 1, \quad y = 0 \dots Y - 1 \quad (2)$$

$$RefV(x, y) \quad x = 0 \dots X - 1, \quad y = 0 \dots Y - 1 \quad (3)$$

where:

X: number of horizontal pixels within a field

Y: number of vertical pixels.

For a *YUV422* input, each *U* and *V* value must be repeated to give the full resolution arrays (2) and (3).

3.2 Crop and offset

This routine crops with offset the degraded input sequence and crops without offset the reference input sequence. The offset parameters *XOffset* and *YOffset* are determined externally and define the number of pixels horizontal and vertical that the degraded sequence is offset from the reference. The picture origin is defined as being in the top left hand corner of the image, with a positive horizontal increment moving right and a positive vertical increment moving down the picture. A value of *XOffset* = 2 indicates that the degraded fields are offset to the right by 2 pixels and a value of *YOffset* = 2 indicates an offset down of 2 pixels. For an input field with *YUV* values stored in *YUV444* format (see § 3.1) in arrays *InYField*, *InUField*, and *InVField* the cropped and offset output is calculated according to (4) to (20).

$$XStart = -XOffset \quad (4)$$

$$\text{if } (XStart < C_x) \quad \text{then } XStart = C_x \quad (5)$$

$$XEnd = X - 1 - XOffset \quad (6)$$

$$\text{if } (XEnd > X - C_x - 1) \quad \text{then } XEnd = X - C_x - 1 \quad (7)$$

$$YStart = -YOffset \quad (8)$$

$$\text{if } (YStart < C_y) \quad \text{then } YStart = C_y \quad (9)$$

$$YEnd = Y - 1 - YOffset \quad (10)$$

$$\text{if } (YEnd > Y - C_y - 1) \quad \text{then } YEnd = Y - C_y - 1 \quad (11)$$

X and Y give the horizontal and vertical field dimensions respectively and C_x and C_y the number of pixels to be cropped from left and right and top and bottom.

For 625 sequences,

$$X=720, \quad Y=288, \quad C_x=30, \quad C_y=10 \quad (12)$$

For 525 sequences,

$$X=720, \quad Y=243, \quad C_x=30, \quad C_y=10 \quad (13)$$

$Xstart$, $Xend$, $Ystart$ and $Yend$ now define the region of each field that will be copied. Pixels outside this region are initialized according to equations (14) to (15), where $YField$, $UField$ and $VField$ are XxY output pixel arrays containing Y , U and V values respectively.

The vertical bars to the left and right of the field are initialized according to:

$$YField(x, y) = 0 \quad x = 0 \dots XStart - 1, XEnd + 1 \dots X - 1 \quad y = 0 \dots Y - 1 \quad (14)$$

$$UField(x, y) = VField(x, y) = 128 \quad x = 0 \dots XStart - 1, XEnd + 1 \dots X - 1 \quad y = 0 \dots Y - 1 \quad (15)$$

The horizontal bars at the top and bottom of the field are initialized according to:

$$YField(x, y) = 0 \quad x = XStart \dots XEnd, \quad y = 0 \dots YStart - 1, YEnd + 1 \dots Y - 1 \quad (16)$$

$$UField(x, y) = VField(x, y) = 128 \quad x = XStart \dots XEnd \quad y = 0 \dots YStart - 1, YEnd + 1 \dots Y - 1 \quad (17)$$

Finally, the pixel values are copied according to:

$$YField(x, y) = InYField(x + XOffset, y + YOffset) \quad x = XStart \dots XEnd \quad y = YStart \dots YEnd \quad (18)$$

$$UField(x, y) = InUField(x + XOffset, y + YOffset) \quad x = XStart \dots XEnd \quad y = YStart \dots YEnd \quad (19)$$

$$VField(x, y) = InVField(x + XOffset, y + YOffset) \quad x = XStart \dots XEnd \quad y = YStart \dots YEnd \quad (20)$$

For the degraded input, cropping and shifting produces output field arrays $DegYField$, $DegUField$ and $DegVField$, whilst cropping without shifting for the reference sequence produces $RefYField$, $RefUField$ and $RefVField$. These XxY two-dimensional arrays are used as inputs to detection routines described below.

3.3 Matching

The matching process produces signals for use within other detection procedures and also detection parameters for use in the integration procedure. The matching signals are generated from a process of finding the best match for small blocks within each degraded field from a buffer of neighbouring reference fields. This process yields a sequence, the matched reference, for use in place of the reference sequence in some of the detection modules.

The matching analysis is performed on 9×9 pixel blocks of the intensity arrays *RefYField* and *DegYField*. Adding a field number dimension to the intensity arrays, pixel (Px, Py) of the reference field N can be represented as:

$$Ref(N, Px, Py) = RefYField(Px, Py) \quad \text{from field } N \quad (21)$$

A 9×9 pixel block with centre pixel (Px, Py) within the N -th field can be represented as:

$$BlockRef(N, Px, Py) = Ref(n, x, y) \quad x = Px - 4 \dots Px + 4, \quad y = Py - 4 \dots Py + 4 \quad (22)$$

$Deg(n, x, y)$ and $BlockDeg(n, x, y)$ can be similarly defined.

For $BlockDeg(N, Px, Py)$, a minimum matching error, $E(N, Px, Py)$, is calculated by searching neighbouring reference fields according to:

$$E(N, Px, Py) = \text{Min}((1/81) \sum_{j=-4}^4 \sum_{k=-4}^4 (Deg(N, Px + j, Py + k) - Ref(n, x + j, y + k))^2) \quad (23)$$

$$n = N - 4, \dots, N + 5$$

$$x = Px - 4, \dots, Px, \dots, Px + 4$$

$$y = Py - 4, \dots, Py, \dots, Py + 4$$

where N is the index of the degraded field containing the degraded block that is being matched.

If equation (23) determines that the best match to $BlockDeg(N, Px, Py)$ is $BlockRef(n_m, x_m, y_m)$ then a matched reference array $MRef$ is updated according to:

$$MRef(N, Px + j, Py + k) = Ref(n_m, x_m + j, y_m + k) \quad j = -4 \dots 4, k = -4 \dots 4 \quad (24)$$

The matching process of first searching for the best match for a degraded block followed by the copying of the resulting block into the matched reference array is repeated for the whole of the desired analysis region. This analysis region is defined by block centre points $Px()$ and $Py()$ according to:

$$Px(h) = 16 + 8 \times h \quad h = 0 \dots Qx - 1 \quad (25)$$

and

$$Py(v) = 16 + 8 \times v \quad v = 0 \dots Qy - 1 \quad (26)$$

where Qx and Qy define the number of horizontal and vertical analysis blocks.

The matching analysis of the N -th field therefore produces a matched reference sequence described by:

$$BlockMRef(N, Px(h), Py(v)) \quad h = 0 \dots Qx - 1, \quad v = 0 \dots Qy - 1 \quad (27)$$

and a set of best match error values:

$$E(N, Px(h), Py(v)) \quad h = 0 \dots Qx - 1, \quad v = 0 \dots Qy - 1 \quad (28)$$

A set of offset arrays $MatT$, $MatX$ and $MatY$ can be defined such that:

$$BlockMRef(N, Px(h), Py(v)) = BlockRef(MatT(h, v), MatX(h, v), MatY(h, v)) \quad (29)$$

$$h = 0 \dots Qx - 1, \quad v = 0 \dots Qy - 1$$

The matching parameters for 625 and 525 broadcast sequences are given in Table 2.

TABLE 2

Search parameters for matching procedure

Parameter	625	525
Q_x	87	87
Q_y	33	28

The analysis region defined by equations (26) and (27) does not cover the complete field size. $MRef$ must therefore be initialized according to equation (29) so that it may be used elsewhere unrestricted.

$$MRef(x, y) = 0 \quad x = 0 \dots X - 1, \quad y = 0 \dots Y - 1 \quad (30)$$

3.3.1 Matching statistics

Horizontal matching statistics from the matching process are calculated for use in the integration process. The best match for each analysis block, determined according to equation (23), is used in the construction of the histogram $histX$ for each field according to:

$$histX(MatX(h, v) - Px(h) + 4) = histX(MatX(h, v) - Px(h) + 4) + 1 \quad (31)$$

$$h = 0 \dots Q_x - 1, \quad v = 0 \dots Q_y - 1$$

where array $histX$ is initialized to zero for each field. The histogram is then used to determine the measure $fXPerCent$ according to:

$$fXPerCent = 100 \times \text{Max}(histX(i)) / \sum_{j=0}^8 histX(j) \quad i = 0 \dots 8 \quad (32)$$

For each field, the $fXPerCent$ measure gives the proportion (%) of matched blocks that contribute to the peak of the matching histogram.

3.3.2 MPSNR

The minimum error, $E()$, for each matched block is used to calculate a matched SNR according to:

$$\text{if } \left(\sum_{h=0}^{Q_x-1} \sum_{v=0}^{Q_y-1} E(N, Px(h), Py(v)) \right) > 0 \quad \text{then} \quad (33)$$

$$MPSNR = 10 \log_{10} \left(Q_x \times Q_y \times 255^2 / \sum_{h=0}^{Q_x-1} \sum_{v=0}^{Q_y-1} E(N, Px(h), Py(v)) \right)$$

$$\text{if } \left(\sum_{h=0}^{Q_x-1} \sum_{v=0}^{Q_y-1} E(N, Px(h), Py(v)) \right) = 0 \quad \text{then } MPSNR = 10 \log_{10}(255^2) \quad (34)$$

3.3.3 Matching vectors

Horizontal, vertical and delay vectors are stored for later use according to:

$$SyncT(h, v) = MatT(h, v) - N \quad h = 0 \dots Q_x - 1, \quad v = 0 \dots Q_y - 1 \quad (35)$$

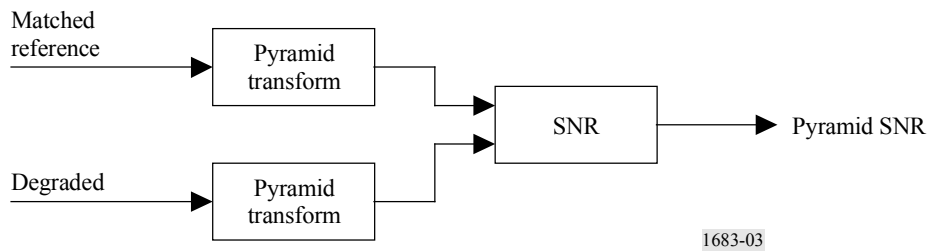
$$SyncX(h, v) = MatX(h, v) - Px(h) \quad h = 0 \dots Q_x - 1, \quad v = 0 \dots Q_y - 1 \quad (36)$$

$$\text{Sync}Y(h,v) = \text{Mat}Y(h,v) - P_y(h) \quad h = 0 \dots Q_x - 1, \quad v = 0 \dots Q_y - 1 \quad (37)$$

3.4 Spatial frequency analysis

The spatial frequency detector is based on a “pyramid” transformation of the degraded and matched reference sequences. First each sequence is transformed to give reference and degraded pyramid arrays. Then, differences between the pyramid arrays are calculated using a mean squared error measure and the results output as a pyramid SNR.

FIGURE 3
Spatial frequency analysis



1683-03

3.4.1 Pyramid transform

Firstly, the input field, F , is copied into a pyramid array, P , according to:

$$P(x,y) = F(x,y) \quad x = 0 \dots X - 1, \quad y = 0 \dots Y - 1 \quad (38)$$

This pyramid array is then updated by three stages ($stage = 0..2$) of horizontal and vertical analysis. The horizontal analysis $Hpy(stage)$ is defined by equations (39) to (43).

First a temporary copy is made of the whole pyramid array:

$$PTemp(x,y) = P(x,y) \quad x = 0 \dots X - 1, \quad y = 0 \dots Y - 1 \quad (39)$$

Then x and y limits are calculated according to:

$$Tx = X / 2^{(stage+1)} \quad (40)$$

$$Ty = Y / 2^{stage} \quad (41)$$

Averages and differences of horizontal pairs of elements of the temporary array are then used to update the pyramid array according to:

$$P(x,y) = 0.5 (PTemp(2x,y) + PTemp(2x+1,y)) \quad x = 0 \dots Tx - 1, \quad y = 0 \dots Ty - 1 \quad (42)$$

$$P(x+Tx,y) = PTemp(2x,y) - PTemp(2x+1,y) \quad x = 0 \dots Tx - 1 \quad y = 0 \dots Ty - 1 \quad (43)$$

The vertical analysis $Vpy(stage)$ is defined by equations (44) to (48).

$$PTemp(x,y) = P(x,y) \quad x = 0 \dots X - 1, \quad y = 0 \dots Y - 1 \quad (44)$$

$$Tx = X / 2^{stage} \quad (45)$$

$$Ty = Y / 2^{(stage+1)} \quad (46)$$

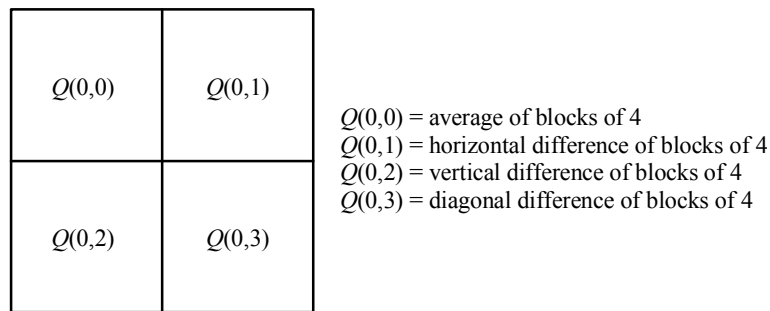
Averages and differences of vertical pairs of elements of the temporary array are then used to update the pyramid array according to:

$$P(x, y) = 0.5 (PTemp(x, 2y) + PTemp(x, 2y + 1)) \quad x = 0 \dots Tx - 1, \quad y = 0 \dots Ty - 1 \quad (47)$$

$$P(x, y + Ty) = PTemp(x, 2y) - PTemp(x, 2y + 1) \quad x = 0 \dots Tx - 1 \quad y = 0 \dots Ty - 1 \quad (48)$$

For stage 0, the horizontal analysis $Hpy(0)$ followed by the vertical analysis $Vpy(0)$ updates the whole of the pyramid array with the 4 quadrants $Q(stage, 0 \dots 3)$ constructed according to:

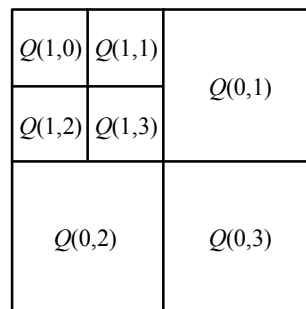
FIGURE 4
Quadrant output from stage 0 analysis



1683-04

Stage 1 analysis is then performed on $Q(0,0)$ to give results $Q(1,0 \dots 3)$ that are stored in the pyramid according to:

FIGURE 5
Quadrant output from stage 1 analysis



1683-05

Stage 2 analysis processes $Q(1,0)$ and overwrites it with $Q(2,0 \dots 3)$.

After the three stages of analysis, the resulting pyramid array has a total of 10 blocks of results. Three blocks $Q(0,1 \dots 3)$ are from the stage $0, 2 \times 2$ pixel analysis, three $Q(1,1 \dots 3)$ from the stage 1, 4×4 analysis and 4 $Q(2,0 \dots 3)$ from the stage 2, 8×8 analysis.

The three-stage analysis of the matched reference and degraded sequences produce the pyramid arrays $Pref$ and $Pdeg$. Differences between these arrays are then measured in the pyramid SNR module.

3.4.2 Pyramid SNR

A squared error measure between the reference and degraded pyramid arrays is determined over quadrants 1 to 3 of stages 0 to 2 according to:

$$E(s, q) = (1/XY^2) \sum_{x=x1(s,q)}^{x2(s,q)-1} \sum_{y=y1(s,q)}^{y2(s,q)-1} (Pref(x, y) - Pdeg(x, y))^2 \quad s = 0...2 \quad q = 1...3 \quad (49)$$

where $x1$, $x2$, $y1$ and $y2$ define the horizontal and vertical limits of the quadrants within the pyramid arrays and are calculated according to:

$$x1(s, 1) = X/2^{(s+1)} \quad x2(s, 1) = 2 \times x1(s, 1) \quad y1(s, 1) = 0 \quad y2(s, 1) = Y/2^{(s+1)} \quad (50)$$

$$x1(s, 2) = 0 \quad x2(s, 2) = X/2^{(s+1)} \quad y1(s, 2) = Y/2^{(s+1)} \quad y2(s, 2) = 2 \times y1(s, 2) \quad (51)$$

$$x1(s, 3) = X/2^{(s+1)} \quad x2(s, 3) = 2 \times x1(s, 3) \quad y1(s, 3) = Y/2^{(s+1)} \quad y2(s, 3) = 2 \times y1(s, 3) \quad (52)$$

The results from equation (49) are then used to determine a PSNR measure for each quadrant of each field according to:

$$\begin{aligned} \text{if } (E > 0, 0) \quad & PySNR(s, q) = 10 \log_{10}(255^2/E(s, q)) \\ \text{then} \quad & SNR = 10 \log_{10}(255^2 \times XY^2) \end{aligned} \quad (53)$$

where the number of stages $s = 0...2$ and the number of quadrants for each stage $q = 1...3$.

3.5 Texture analysis

The texture of the degraded sequence is measured by recording the number of turning-points in the intensity signal along horizontal picture lines. This may be calculated according to equations (54) to (59).

For each field, first a turning-point counter is initialized according to equation (54).

$$sum = 0 \quad (54)$$

Then, each line, $y = 0...Y - 1$, is processed for $x = 0...X - 2$ according:

$$last_pos = 0, \quad last_neg = 0 \quad (55)$$

$$dif(x) = P(x, y) - P(x + 1, y) \quad (56)$$

$$\text{if } ((dif(x) < 0) \text{ AND } (last_neg < last_pos)) sum = sum + 1 \quad (57)$$

$$\text{if } ((dif(x) > 0) \text{ AND } (last_neg > last_pos)) sum = sum + 1 \quad (58)$$

$$\text{if } (dif(x) > 0) last_pos = x \quad (59)$$

$$\text{if } (dif(x) < 0) last_neg = x \quad (60)$$

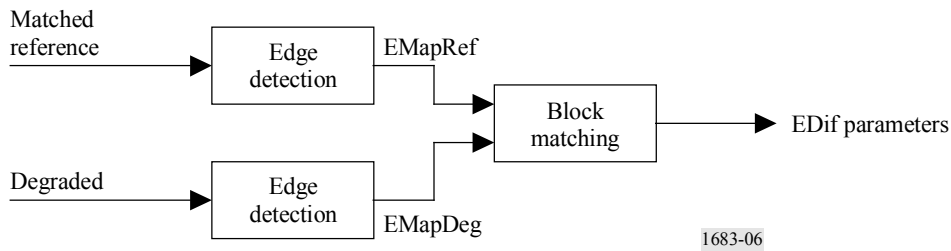
When all the lines for a field have been processed, the counter, sum , will contain the number of turning-points in the horizontal intensity signal. This is then used to calculate a texture parameter for each field according to:

$$TextureDeg = sum \times 100 / XY \quad (61)$$

3.6 Edge analysis

Each field of the degraded and matched reference sequences is separately passed through an edge detection routine to produce corresponding edge field maps, which are then compared in a block matching procedure to produce the detection parameters.

FIGURE 6
Edge analysis



3.6.1 Edge detection

A Canny edge detector [Canny, 1986] was used to determine the edge maps, but other similar edge detection techniques may be used. The resulting edge maps, *EMapRef* and *EMapDeg*, are pixel maps with an edge indicated by a 1 and no edge by 0.

For an edge detected at pixel (x, y) :

$$EMap(x, y) = 1 \quad x = 0 \dots X - 1, \quad y = 0 \dots Y - 1 \quad (62)$$

For no edge detected at pixel (x, y) :

$$EMap(x, y) = 0 \quad x = 0 \dots X - 1, \quad y = 0 \dots Y - 1 \quad (63)$$

3.6.2 Edge differencing

The edge differencing procedure measures the differences between the edge maps for corresponding degraded and matched reference fields. The analysis is performed in $N \times M$ pixel non-overlapping blocks according to equations (64) to (68).

First, a measure of the number of edge-marked pixels in each analysis block is calculated, where B_h and B_v define the number of non-overlapping blocks to be analysed in the horizontal and vertical directions and X_1 and Y_1 define analysis offsets from the field edge.

$$Bref(x, y) = \sum_{i=1}^{i2} \sum_{j=1}^{j2} EMapRef(Nx + X_1 + i, My + Y_1 + j) \quad x = 0 \dots B_h - 1, y = 0 \dots B_v - 1 \quad (64)$$

$$BDeg(x, y) = \sum_{i=1}^{i2} \sum_{j=1}^{j2} EMapDeg(Nx + X_1 + i, My + Y_1 + j) \quad x = 0 \dots B_h - 1, y = 0 \dots B_v - 1 \quad (65)$$

The summation limits are determined according to:

$$i1 = -(N \text{ div } 2) \quad i2 = (N - 1) \text{ div } 2 \quad (66)$$

$$j1 = -(M \text{ div } 2) \quad j2 = (M - 1) \text{ div } 2 \quad (67)$$

where the “div” operator represents an integer division.

Then, a measure of the differences over the whole field is calculated according to:

$$EDif = (1/(N M Bh Bv)) \left(\sum_{x=0}^{Bh-1} \sum_{y=0}^{Bv-1} (BRef(x, y) - BDeg(x, y))^Q \right)^{1/Q} \quad (68)$$

For 720×288 pixel fields for 625 broadcast video:

$$N = 4, X1 = 6, Bh = 178 \quad M = 4, Y1 = 10, Bv = 69, Q = 3 \quad (69)$$

For 720×243 pixel fields for 525 broadcast video:

$$N = 4, X1 = 6, Bh = 178 \quad M = 4, Y1 = 10, Bv = 58, Q = 3 \quad (70)$$

3.7 MPSNR analysis

A matched signal to noise ratio is calculated for the pixel V values by use of the matching vectors defined in equations (35) to (37). For each set of matching vectors, an error measure, VE , is calculated according to:

$$VE(h, v) = (1/81) \sum_{i=-4}^4 \sum_{j=-4}^4 (DegV(N, Px(h) + i, Py(h) + j) - RefVField(N + SyncT(h, v), Px(h) + SyncX(h, v) + i, Py(v) + SyncY(h, v) + j))^2 \quad (71)$$

A segmental PSNR measure is then calculated for the field according to:

$$SegVPSNR = (1/Qx Qy) \sum_{h=0}^{Qx-1} \sum_{v=0}^{Qy-1} 10 \log_{10}(255^2 / (VE(h, v) + 1)) \quad (72)$$

4 Integration

The integration procedure firstly requires the time averaging of the field-by-field detection parameters according to equation (73):

$$AvD(k) = (1/N) \sum_{n=0}^{N-1} D(k, n) \quad k = 0 \dots 5 \quad (73)$$

where:

N : total number of fields in the tested sequences

$D(k, n)$: detection parameter k for field n .

The averaged detection parameters, $AvD(k)$, are then combined to give a predicted quality score, PDMOS, for the N field sequence according to equation (74):

$$PDMOS = Offset + \sum_{k=0}^5 AvD(k) \times W(k) \quad (74)$$

Tables 3 and 4 show the integrator parameters for 625 and 525 sequences respectively.

TABLE 3

Integration parameters for 625 broadcast video

<i>K</i>	Parameter name	<i>W</i>
0	<i>TextureDeg</i>	−0.68
1	<i>PySNR</i> (3,3)	−0.57
2	<i>EDif</i>	+58 913.294
3	<i>fXPerCent</i>	−0.208
4	<i>MPSNR</i>	−0.928
5	<i>SegVPSNR</i>	−1.529
Offset	+176.486	
<i>N</i>	400	

TABLE 4

Integration parameters for 525 broadcast video

<i>K</i>	Parameter name	<i>W</i>
0	<i>TextureDeg</i>	+0.043
1	<i>PySNR</i> (3,3)	−2.118
2	<i>EDif</i>	+60 865.164
3	<i>fXPerCent</i>	−0.361
4	<i>MPSNR</i>	+1.104
5	<i>SegVPSNR</i>	−1.264
Offset	+260.773	
<i>N</i>	480	

5 Registration

The FR model requires both spatial and temporal alignment to operate effectively. The model incorporates inherent alignment and can accommodate spatial offsets between the reference and degraded sequences ± 4 pixels and temporal offset of ± 4 fields. Spatial and temporal offsets beyond these limits are not handled by the model and a separate registration module will be required to ensure the reference and degraded files are properly aligned.

6 References

CANNY, J. [1986] A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*. Vol. 8(6), p. 679-698.

Copyright International Telecommunication Union/ITU Radiocommunication Sector

Annex 2a

TABLE 5
525 subjective and objective data

Filename	Source sequence (SRC)	Hypothetical reference circuit (HRC)	Raw mean subjective rating	Model predicted rating based on raw data	Scaled mean subjective rating	Model predicted rating based on scaled data
V2src01_hrc01_525.yuv	1	1	-38.30757576	44.945049	0.5402368	0.69526
V2src01_hrc02_525.yuv	1	2	-39.56212121	38.646271	0.5483205	0.58989
V2src01_hrc03_525.yuv	1	3	-25.9469697	32.855755	0.4024097	0.50419
V2src01_hrc04_525.yuv	1	4	-17.24090909	21.062775	0.3063528	0.36089
V2src02_hrc01_525.yuv	2	1	-35.23636364	31.260744	0.5025558	0.48242
V2src02_hrc02_525.yuv	2	2	-18.01818182	18.732758	0.3113346	0.33715
V2src02_hrc03_525.yuv	2	3	-6.284848485	8.914509	0.1881739	0.25161
V2src02_hrc04_525.yuv	2	4	-6.983333333	4.16663	0.1907347	0.21776
V2src03_hrc01_525.yuv	3	1	-31.96515152	22.348713	0.4682724	0.37461
V2src03_hrc02_525.yuv	3	2	-17.47727273	10.44728	0.3088831	0.26352
V2src03_hrc03_525.yuv	3	3	-1.104545455	2.494911	0.1300389	0.20688
V2src03_hrc04_525.yuv	3	4	-1.171212121	0	0.1293293	0.19158
V2src04_hrc05_525.yuv	4	5	-50.64090909	40.82526	0.6742005	0.6249
V2src04_hrc06_525.yuv	4	6	-28.05454545	32.552322	0.4250873	0.49999
V2src04_hrc07_525.yuv	4	7	-23.87575758	25.286598	0.3762656	0.40764
V2src04_hrc08_525.yuv	4	8	-16.60757576	19.86405	0.2972294	0.3485
V2src05_hrc05_525.yuv	5	5	-31.86969697	30.812616	0.4682559	0.47645
V2src05_hrc06_525.yuv	5	6	-18.56515152	21.413895	0.3203024	0.3646
V2src05_hrc07_525.yuv	5	7	-8.154545455	15.446437	0.2071702	0.306
V2src05_hrc08_525.yuv	5	8	-4.006060606	10.836051	0.1652752	0.26662
V2src06_hrc05_525.yuv	6	5	-41.63181818	37.342789	0.5690291	0.56967
V2src06_hrc06_525.yuv	6	6	-29.48787879	26.660055	0.4370961	0.42391
V2src06_hrc07_525.yuv	6	7	-22.25909091	20.878248	0.3591788	0.35896
V2src06_hrc08_525.yuv	6	8	-12.03181818	16.896168	0.2482169	0.31941
V2src07_hrc05_525.yuv	7	5	-23.89545455	19.086998	0.3796362	0.34067
V2src07_hrc06_525.yuv	7	6	-10.15606061	10.69402	0.2276934	0.26548
V2src07_hrc07_525.yuv	7	7	-4.240909091	4.896546	0.1644409	0.22267
V2src07_hrc08_525.yuv	7	8	-5.98030303	1.555055	0.1819566	0.20099
V2src08_hrc09_525.yuv	8	9	-76.2	52.094177	0.9513387	0.83024
V2src08_hrc10_525.yuv	8	10	-61.34545455	47.395226	0.789748	0.7397
V2src08_hrc11_525.yuv	8	11	-66.02575758	52.457584	0.8405916	0.83753

TABLE 5
525 subjective and objective data

Filename	Source sequence (SRC)	Hypothetical reference circuit (HRC)	Raw mean subjective rating	Model predicted rating based on raw data	Scaled mean subjective rating	Model predicted rating based on scaled data
V2src08_hrc12_525.yuv	8	12	-37.20454545	37.931854	0.5221555	0.57874
V2src08_hrc13_525.yuv	8	13	-31.23030303	30.95985	0.4572049	0.4784
V2src08_hrc14_525.yuv	8	14	-31.26818182	33.293602	0.4614104	0.51031
V2src09_hrc09_525.yuv	9	9	-64.42878788	54.414772	0.8262912	0.87746
V2src09_hrc10_525.yuv	9	10	-49.92878788	36.080425	0.660339	0.55061
V2src09_hrc11_525.yuv	9	11	-53.73181818	46.338791	0.7100111	0.72031
V2src09_hrc12_525.yuv	9	12	-34.36969697	23.21393	0.4921708	0.38409
V2src09_hrc13_525.yuv	9	13	-22.85454545	16.955978	0.3656559	0.31998
V2src09_hrc14_525.yuv	9	14	-16.41666667	13.694396	0.2960957	0.29046
V2src10_hrc09_525.yuv	10	9	-72.11212121	48.179104	0.9084171	0.75433
V2src10_hrc10_525.yuv	10	10	-43.11666667	30.703861	0.5908784	0.475
V2src10_hrc11_525.yuv	10	11	-56.11969697	52.63887	0.7302376	0.84118
V2src10_hrc12_525.yuv	10	12	-19.55909091	21.95225	0.3345703	0.37033
V2src10_hrc13_525.yuv	10	13	-12.34393939	16.23988	0.2565459	0.31328
V2src10_hrc14_525.yuv	10	14	-16.05	23.201355	0.2953144	0.38395
V2src11_hrc09_525.yuv	11	9	-50.40454545	36.394535	0.6675853	0.55531
V2src11_hrc10_525.yuv	11	10	-54.26212121	37.812542	0.7054929	0.5769
V2src11_hrc11_525.yuv	11	11	-41.73636364	44.128036	0.5761193	0.68087
V2src11_hrc12_525.yuv	11	12	-19.03939394	14.619688	0.32761	0.29857
V2src11_hrc13_525.yuv	11	13	-17.72121212	14.12041	0.310495	0.29417
V2src11_hrc14_525.yuv	11	14	-19.4969697	14.927424	0.331051	0.30132
V2src12_hrc09_525.yuv	12	9	-61.35	40.051254	0.7883371	0.61229
V2src12_hrc10_525.yuv	12	10	-46.84545455	31.128973	0.6295301	0.48066
V2src12_hrc11_525.yuv	12	11	-51.80151515	41.77285	0.6809288	0.6406
V2src12_hrc12_525.yuv	12	12	-22.51969697	20.868282	0.3651402	0.35886
V2src12_hrc13_525.yuv	12	13	-14.17878788	15.040992	0.2714356	0.30234
V2src12_hrc14_525.yuv	12	14	-14.6030303	13.521517	0.2782449	0.28896
V2src13_hrc09_525.yuv	13	9	-55.25	38.691498	0.7211194	0.5906
V2src13_hrc10_525.yuv	13	10	-39.55	33.054504	0.5545722	0.50696
V2src13_hrc11_525.yuv	13	11	-40.03939394	45.9454	0.5525494	0.71318
V2src13_hrc12_525.yuv	13	12	-14	16.631002	0.2708744	0.31692
V2src13_hrc13_525.yuv	13	13	-14.33181818	15.113959	0.27549	0.30299
V2src13_hrc14_525.yuv	13	14	-14.31969697	16.611286	0.2733771	0.31674



TABLE 6
625 subjective and objective data

Filename	SRC	HRC	Raw mean subjective rating	Model predicted rating based on raw data	Scaled mean subjective rating	Model predicted rating based on scaled data
V2src1_hrc2_625.yuv	1	2	38.85185185	31.764214	0.59461	0.47326
V2src1_hrc3_625.yuv	1	3	42.07407407	21.868561	0.64436	0.36062
V2src1_hrc4_625.yuv	1	4	23.77777778	12.195552	0.40804	0.27239
V2src1_hrc6_625.yuv	1	6	18.14814815	9.169512	0.34109	0.24887
V2src1_hrc8_625.yuv	1	8	12.92592593	6.738072	0.2677	0.23128
V2src1_hrc10_625.yuv	1	10	11.88888889	2.553883	0.26878	0.20356
V2src2_hrc2_625.yuv	2	2	33.51851852	31.492788	0.54173	0.46985
V2src2_hrc3_625.yuv	2	3	46.48148148	31.1313	0.70995	0.46535
V2src2_hrc4_625.yuv	2	4	13.33333333	20.241726	0.27443	0.34432
V2src2_hrc6_625.yuv	2	6	8.814814815	17.39045	0.22715	0.31721
V2src2_hrc8_625.yuv	2	8	7.074074074	14.914576	0.21133	0.29513
V2src2_hrc10_625.yuv	2	10	3.407407407	7.352309	0.16647	0.23562
V2src3_hrc2_625.yuv	3	2	48.07407407	38.852715	0.73314	0.56845
V2src3_hrc3_625.yuv	3	3	50.66666667	38.244621	0.76167	0.55982
V2src3_hrc4_625.yuv	3	4	32.11111111	27.733229	0.49848	0.42454
V2src3_hrc6_625.yuv	3	6	22.33333333	24.80323	0.38613	0.39159
V2src3_hrc8_625.yuv	3	8	16.33333333	23.296747	0.34574	0.37544
V2src3_hrc10_625.yuv	3	10	11.96296296	16.33028	0.26701	0.30759
V2src4_hrc2_625.yuv	4	2	36.14814815	42.041592	0.58528	0.61514
V2src4_hrc3_625.yuv	4	3	55.03703704	49.283836	0.90446	0.72942
V2src4_hrc4_625.yuv	4	4	39.7037037	38.322186	0.62361	0.56091
V2src4_hrc6_625.yuv	4	6	38.03703704	36.863457	0.61143	0.54053
V2src4_hrc8_625.yuv	4	8	24.40740741	32.46579	0.43329	0.48214
V2src4_hrc10_625.yuv	4	10	12.88888889	25.918123	0.26548	0.40388
V2src5_hrc2_625.yuv	5	2	38.62962963	38.95779	0.61973	0.56995
V2src5_hrc3_625.yuv	5	3	44.18518519	40.076313	0.68987	0.58609
V2src5_hrc4_625.yuv	5	4	24.66666667	23.166002	0.41648	0.37406
V2src5_hrc6_625.yuv	5	6	23.62962963	20.592213	0.4218	0.34778
V2src5_hrc8_625.yuv	5	8	12.40740741	13.763152	0.27543	0.28531
V2src5_hrc10_625.yuv	5	10	7.37037037	8.418313	0.2022	0.24332
V2src6_hrc2_625.yuv	6	2	22.48148148	33.810165	0.38852	0.49949
V2src6_hrc3_625.yuv	6	3	27.07407407	25.004984	0.44457	0.39379
V2src6_hrc4_625.yuv	6	4	13.18518519	20.889347	0.27983	0.35074
V2src6_hrc6_625.yuv	6	6	14.44444444	17.418222	0.28106	0.31747



TABLE 6
625 subjective and objective data

Filename	SRC	HRC	Raw mean subjective rating	Model predicted rating based on raw data	Scaled mean subjective rating	Model predicted rating based on scaled data
V2src6_hrc8_625.yuv	6	8	8.740740741	15.486559	0.23726	0.30011
V2src6_hrc10_625.yuv	6	10	5.518518519	11.509192	0.17793	0.2669
V2src7_hrc4_625.yuv	7	4	39.25925926	45.231079	0.59953	0.66412
V2src7_hrc6_625.yuv	7	6	33.85185185	43.131519	0.55093	0.63163
V2src7_hrc9_625.yuv	7	9	27.07407407	39.506535	0.45163	0.57784
V2src7_hrc10_625.yuv	7	10	19.25925926	34.418381	0.35617	0.50749
V2src8_hrc4_625.yuv	8	4	15.85185185	40.408993	0.32528	0.59095
V2src8_hrc6_625.yuv	8	6	17.03703704	38.552574	0.32727	0.56418
V2src8_hrc9_625.yuv	8	9	14.85185185	35.577034	0.30303	0.52297
V2src8_hrc10_625.yuv	8	10	11.48148148	30.278536	0.26366	0.45484
V2src9_hrc4_625.yuv	9	4	28.96296296	30.515778	0.47656	0.45775
V2src9_hrc6_625.yuv	9	6	30.51851852	26.971027	0.49924	0.41577
V2src9_hrc9_625.yuv	9	9	19.66666667	23.351355	0.39101	0.37601
V2src9_hrc10_625.yuv	9	10	20.92592593	17.856861	0.37122	0.32152
V2src10_hrc4_625.yuv	10	4	40.33333333	43.640377	0.70492	0.63942
V2src10_hrc6_625.yuv	10	6	37.33333333	40.552502	0.58218	0.59305
V2src10_hrc9_625.yuv	10	9	30.92592593	36.747391	0.49711	0.53893
V2src10_hrc10_625.yuv	10	10	21.2962963	30.161013	0.37854	0.45341
V2src11_hrc1_625.yuv	11	1	50.25925926	55.909908	0.79919	0.84263
V2src11_hrc5_625.yuv	11	5	35.51851852	44.049999	0.59256	0.64572
V2src11_hrc7_625.yuv	11	7	18.7037037	26.877754	0.34337	0.4147
V2src11_hrc10_625.yuv	11	10	15.07407407	23.420477	0.30567	0.37674
V2src12_hrc1_625.yuv	12	1	36.33333333	43.837097	0.61418	0.64244
V2src12_hrc5_625.yuv	12	5	38.44444444	40.349903	0.6661	0.59008
V2src12_hrc7_625.yuv	12	7	31.11111111	37.254383	0.53242	0.54594
V2src12_hrc10_625.yuv	12	10	26.14814815	28.953564	0.44737	0.43887
V2src13_hrc1_625.yuv	13	1	43.7037037	38.333649	0.74225	0.56108
V2src13_hrc5_625.yuv	13	5	43.2962963	34.290554	0.66799	0.5058
V2src13_hrc7_625.yuv	13	7	25.2962963	26.990025	0.42065	0.41598
V2src13_hrc10_625.yuv	13	10	15.88888889	20.181463	0.33381	0.34373

Annex 3

Model 2

CONTENTS

	<i>Page</i>
1 Introduction	20
2 Objective measurement of video quality based on edge degradation	21
2.1 Edge PSNR (EPSNR)	21
2.2 Post adjustments	28
2.2.1 De-emphasis of high EPSNR	28
2.2.2 Considering blurred edges	28
2.2.3 Scaling	29
2.3 Registration accuracy	29
2.4 The block diagram of the model	29
3 Objective data	29
4 Conclusion	29
5 References	29

1 Introduction

Traditionally, the evaluation of video quality is performed by a number of evaluators who subjectively evaluate the video quality. The evaluation can be done with or without reference videos. In referenced evaluation, evaluators are shown two videos: the reference (source) video and the processed video that is to be compared with the source video. By comparing the two videos, the evaluators give subjective scores to the videos. Therefore, it is often called a subjective test of video quality. Although the subjective test is considered to be the most accurate method since it reflects human perception, it has several limitations. First of all, it requires a number of evaluators. Thus, it is time-consuming and expensive. Furthermore, it cannot be done in real time. As a result, there has been a great interest in developing objective methods for video quality measurement. An important requirement for an objective method for video quality measurement is that it should provide consistent performance results over a wide range of video sequences that are not used in the design stage. Toward this goal, a model was developed, which is easy to implement, fast enough for real-time implementations and robust over a wide range of video impairments. The model is a product of collaborative works from Yonsei University, SK Telecom, and Radio Research Laboratory, Republic of Korea.

2 Objective measurement of video quality based on edge degradation

2.1 Edge PSNR (EPSNR)

The model for objective video quality measurement is a full reference method. In other words, it is assumed that a reference video is provided. By analysing how humans perceive video quality, it is observed that the human visual system is sensitive to degradation around the edges. In other words, when the edge areas of a video are blurred, evaluators tend to give low scores to the video even though the overall mean squared error is small. 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 EPSNR is computed and used as a video quality metric after post-processing.

In the model, an edge detection algorithm needs to be first applied to locate edge areas. 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 areas. 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 areas. In other words, pixels whose magnitude gradients exceed a threshold value are considered as edge areas.

Figures 7-11 illustrate the above procedure. Figure 7 shows a source image. Figure 8 shows a horizontal gradient image $g_{horizontal}(m,n)$, which is obtained by applying a horizontal gradient operator to the source image of Fig. 7. Figure 9 shows a vertical gradient image $g_{vertical}(m,n)$, which is obtained by applying a vertical gradient operator to the source image of Fig. 7. Figure 10 shows the magnitude gradient image (edge image) and Fig. 11 shows the binary edge image (mask image) obtained by applying thresholding to the magnitude gradient image of Fig. 10.

FIGURE 7

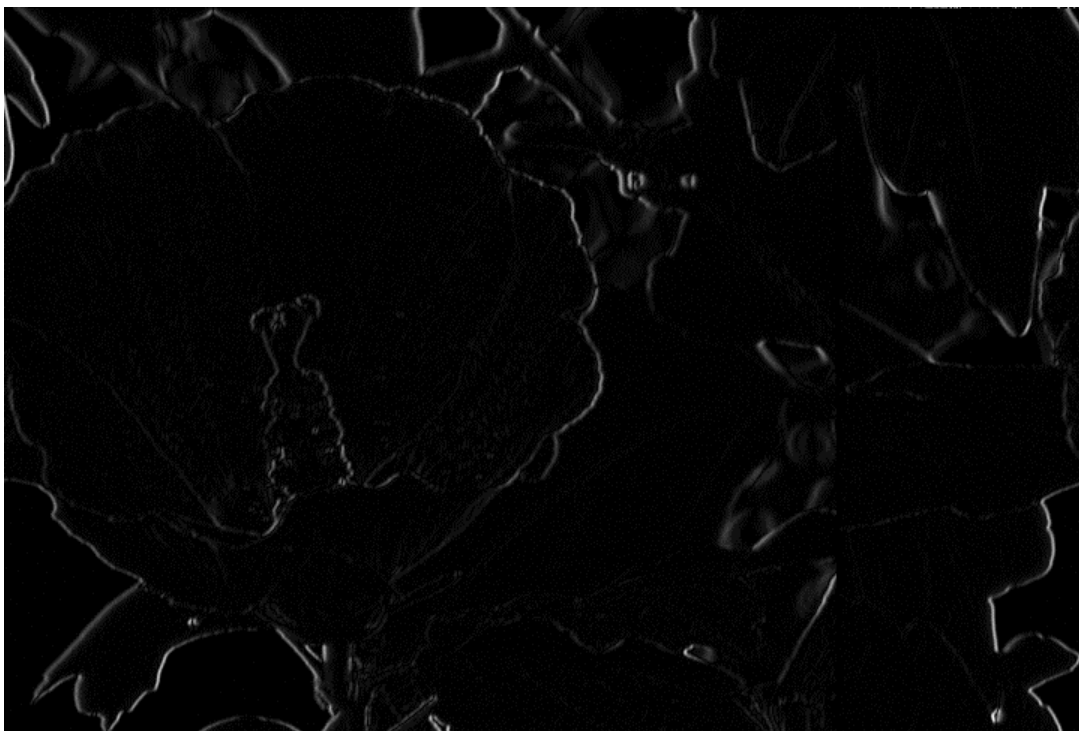
A source image (original image)



1683-07

FIGURE 8

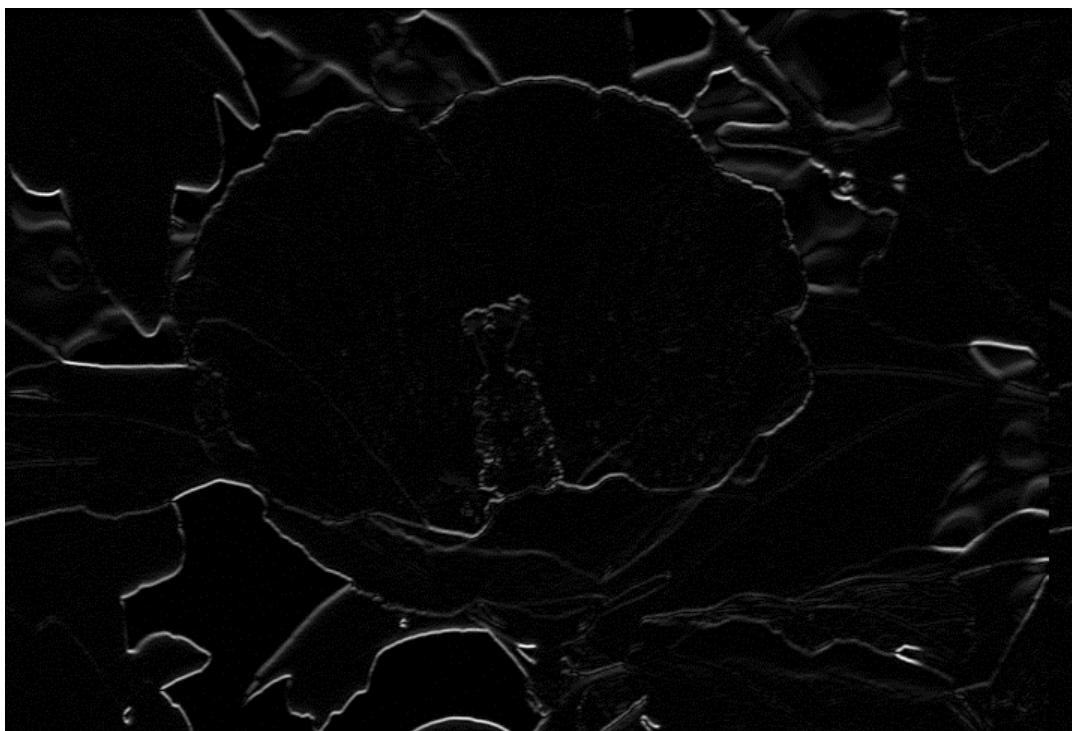
A horizontal gradient image, which is obtained by applying a horizontal gradient operator to the source image of Fig. 7



1683-08

FIGURE 9

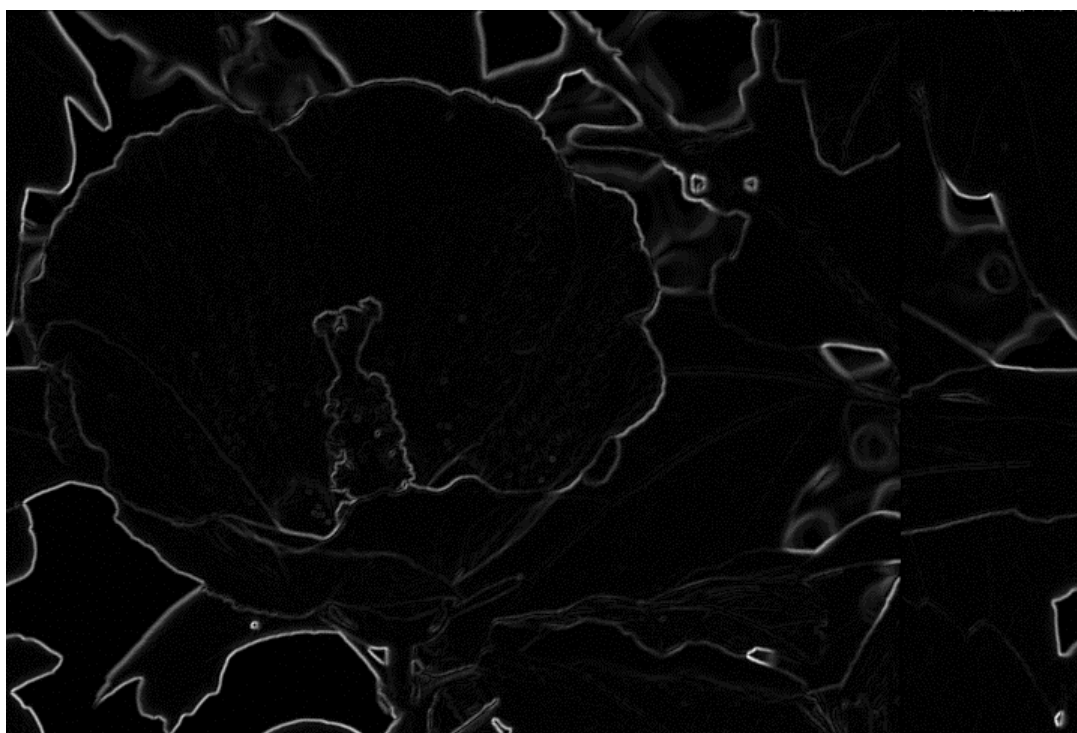
A vertical gradient image, which is obtained by applying a vertical gradient operator to the source image of Fig. 7



1683-09

FIGURE 10

A magnitude gradient image



1683-10

FIGURE 11

A binary edge image (mask image) obtained by applying thresholding to the magnitude gradient image of Fig. 10

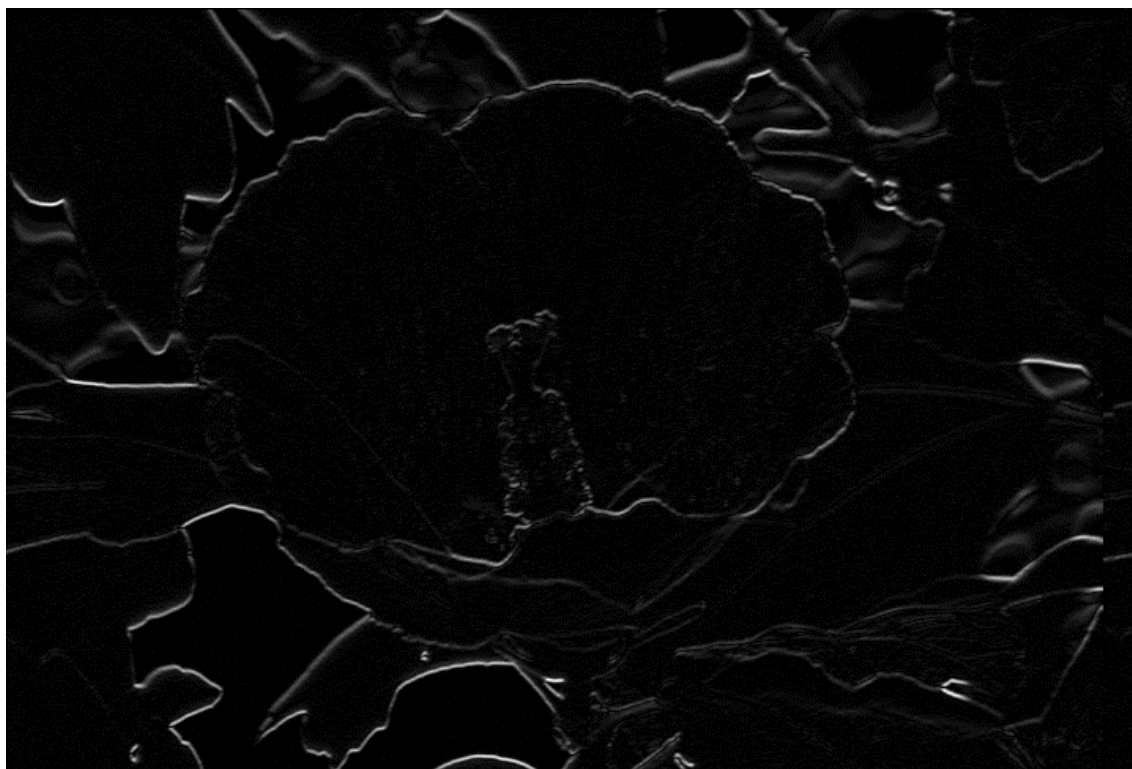


1683-11

Alternatively, one may use a modified procedure to find edge areas. 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 areas. In other words, pixels of the modified successive gradient image, which exceed a threshold value, are considered as edge areas. Figures 12-15 illustrate the modified procedure. Figure 12 shows a vertical gradient image $g_{vertical}(m,n)$, which is obtained by applying a vertical gradient operator to the source image of Fig. 7. Figure 13 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 Fig. 12. Figure 14 shows the binary edge image (mask image) obtained by applying thresholding to the modified successive gradient image of Fig. 13.

FIGURE 12

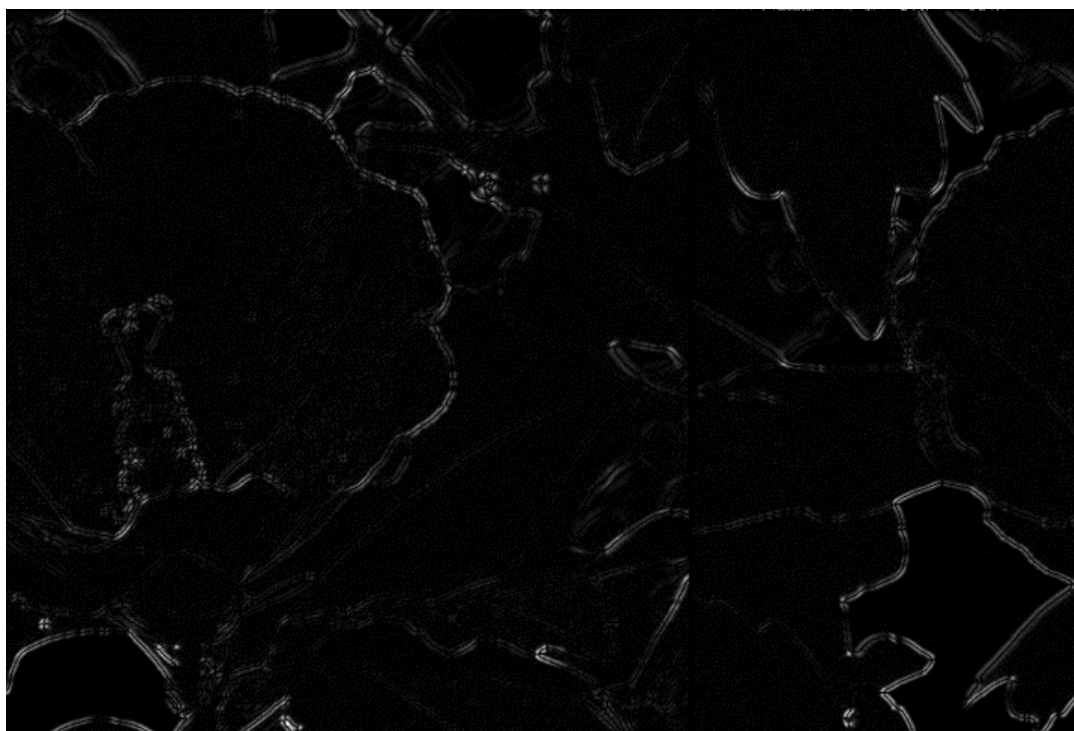
A vertical gradient image, which is obtained by applying a vertical gradient operator to the source image of Fig. 7



1683-12

FIGURE 13

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 Fig. 12



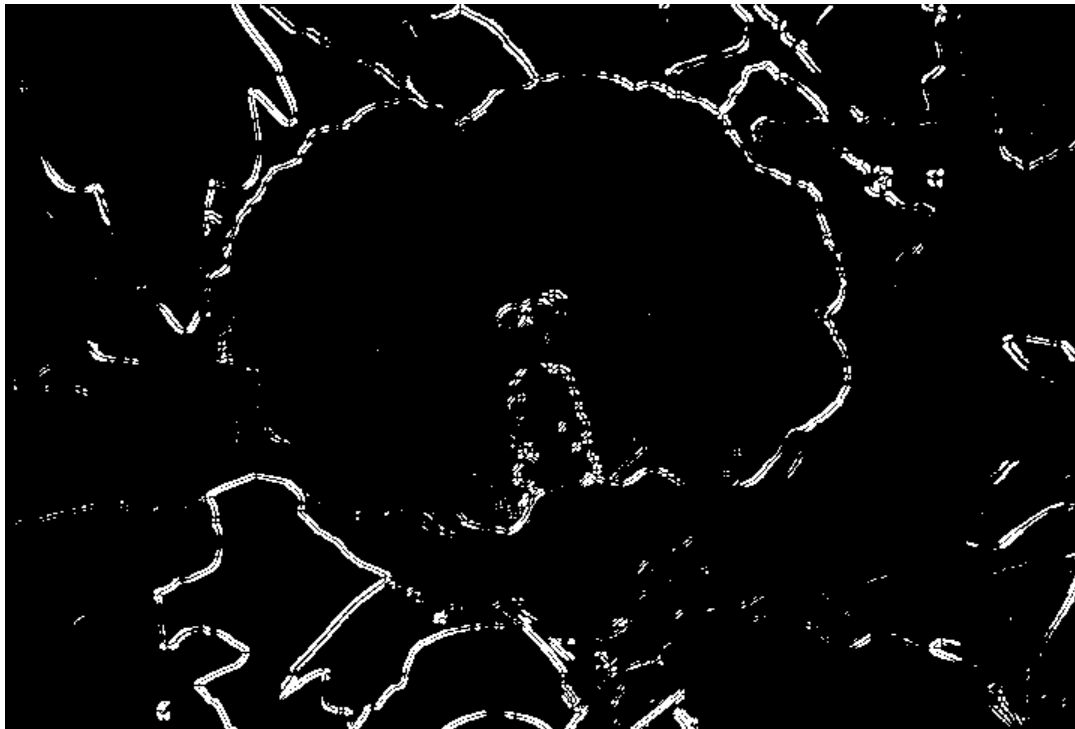
1683-13

**Documents**

Document Sharing Hub

FIGURE 14

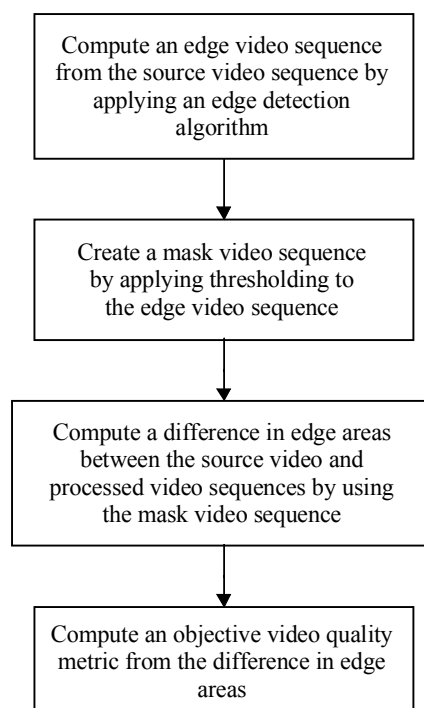
A binary edge image (mask image) obtained by applying thresholding to the modified successive gradient image of Fig. 13



1683-14

FIGURE 15

A block diagram of EPSNR



1683-15

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 videos and compression algorithms. However, some methods may outperform other methods.

Thus, in the model, an edge detection operator is first applied, producing edge images (Figs. 10 and 13). Then, a mask image (binary edge image) is produced by applying thresholding to the edge image (Figs. 11 and 14). 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 11 and 14 show examples of mask images. It is noted that this edge detection algorithm is applied to the source image. Although one may apply the edge detection algorithm to processed images, it is more accurate to apply it to the source 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.

Next, differences between the source video sequence and processed video sequence, corresponding to non-zero pixels of the mask image are computed. In other words, the squared error of edge areas of the l -th frame is computed as follows:

$$se_e^l = \sum_{i=1}^M \sum_{j=1}^N \{S^l(i, j) - P^l(i, j)\}^2 \quad \text{if } |R^l(i, j)| \neq 0 \quad (75)$$

where:

- $S^l(i, j)$: l -th image of the source video sequence
- $P^l(i, j)$: l -th image of the processed video sequence
- $R^l(i, j)$: l -th image of the mask video sequence
- M : number of rows
- N : number of columns.

When the model is implemented, one may skip the generation of the mask video sequence. In fact, without creating the mask video sequence, the squared error of edge areas of the l -th frame is computed as follows:

$$se_e^l = \sum_{i=1}^M \sum_{j=1}^N \{S^l(i, j) - P^l(i, j)\}^2 \quad \text{if } |Q^l(i, j)| \geq t_e \quad (76)$$

where:

- $Q^l(i, j)$: l -th image of the edge video sequence
- t_e : threshold.

Although the mean squared error is used in equation (75) to compute the difference between the source video sequence and the processed video sequence, any other type of difference may be used. For instance, the absolute difference may be also used. In the model submitted to the VQEG Phase II test, t_e , was set to 260 and the modified edge detection algorithm was used with the Sobel operator.

This procedure is repeated for the entire video sequences and the edge mean squared error is computed as follows:

$$mse_e = \frac{1}{K} \sum_{l=1}^L se_e^l \quad (77)$$

where:

L : number of images (frames or fields)

K : total number of pixels of the edge areas.

Finally, the PSNR of the edge areas (EPNSR) is computed as follows:

$$EPSNR = 10 \log_{10} \left(\frac{P^2}{mse_e} \right) \quad (78)$$

where:

P : peak pixel value.

In the model, this EPSNR is used as a basic objective video quality score. Figure 15 shows a block diagram of computing the EPSNR.

2.2 Post adjustments

2.2.1 De-emphasis of high EPSNR

When the value of EPSNR is over 35, it appears that the EPSNR overestimates perceptual quality. Thus, the following piecewise linear scaling is used:

$$EPSNR = \begin{cases} EPSNR & \text{if } 0 \leq EPSNR \leq 35 \\ EPSNR \times 0.9 & \text{if } 35 < EPSNR \leq 40 \\ EPSNR \times 0.8 & \text{if } EPSNR > 40 \end{cases} \quad (79)$$

2.2.2 Considering blurred edges

It is observed that when edges are severely blurred in low quality videos, evaluators tend to give lower subjective scores. In other words, if the edge areas of the processed video sequence are substantially smaller than those of the source video sequence, the evaluators give lower scores. Furthermore, it is observed that some video sequences have a very small number of pixels which have high frequency components. In other words, the number of pixels of edge areas is very small. In order to take into account these problems, the edge areas of the source and processed video sequences are computed and the EPSNR is modified as follows:

$$MEPSNR = \begin{cases} MEPSNR - 60 \times \left(0.1.225 - \left(\frac{EP_{common}}{EP_{src}} \right)^2 \right) \\ MEPSNR \end{cases}$$

if $EPSNR < 25$ and $\left(\frac{EP_{common}}{EP_{src}} \right)^2 < 0.35$ (80)

and $\frac{EP_{hrc}}{EP_{src}} < 0.13$ elsewhere

where:

$MEPSNR$: modified EPSNR

EP_{common} : total number of common edge pixels in the SRC and HRC video sequences (i.e. edge pixels occurring at the same location)

EP_{src} : total number of edge pixels in the SRC (source) video sequence.

For some video sequences, EP_{src} can be very small. If EP_{src} is smaller than 10 000 pixels (about $10\,000/240 = 41.7$ pixels per frame for the 8 s 525 videos and about $10\,000/200 = 50$ pixels per frame for the 8 s 625 videos), the user may reduce threshold t_e in equation (76) by 20 until EP_{src} is larger than or equal to 10 000 pixels. If EP_{src} is smaller than 10 000 pixels even when t_e is reduced to 80, the post adjustment using equation (80) is not used. In this case, the $EPSNR$ is computed using $t_e = 60$. If this option is taken, the user may delete the condition of $EP_{hrc}/EP_{src} < 0.13$ in equation (80).

2.2.3 Scaling

Next, scale objective scores are rescaled so that they will be between 0 (not distinguishable from the original video) and 1.

$$VQM = 1 - MEPSNR \times 0.02 \quad (81)$$

This VQM is used as the objective score of the model.

2.3 Registration accuracy

The recommended registration accuracy for the model is a half-pixel accuracy in the interlaced videos, which is equivalent to a quarter-pixel accuracy in the progressive video format. The cubic spline interpolation [Lee *et al.*, 1998] or better is strongly recommended to calculate sub-pixel values.

2.4 The block diagram of the model

Figure 16 shows the complete block diagram of the model.

3 Objective data

The model was applied to the VQEG Phase II video data¹. However, after the model was submitted, registration and operator errors were found. The objective data presented in this Annex is the same data as in the VQEG Phase II Final Report. Consequently, when the method described in this Annex is properly implemented, the user may obtain different objective data from those of this Annex. Tables 7 to 8 show the objective data of the 525 and 625 video data sets.

4 Conclusion

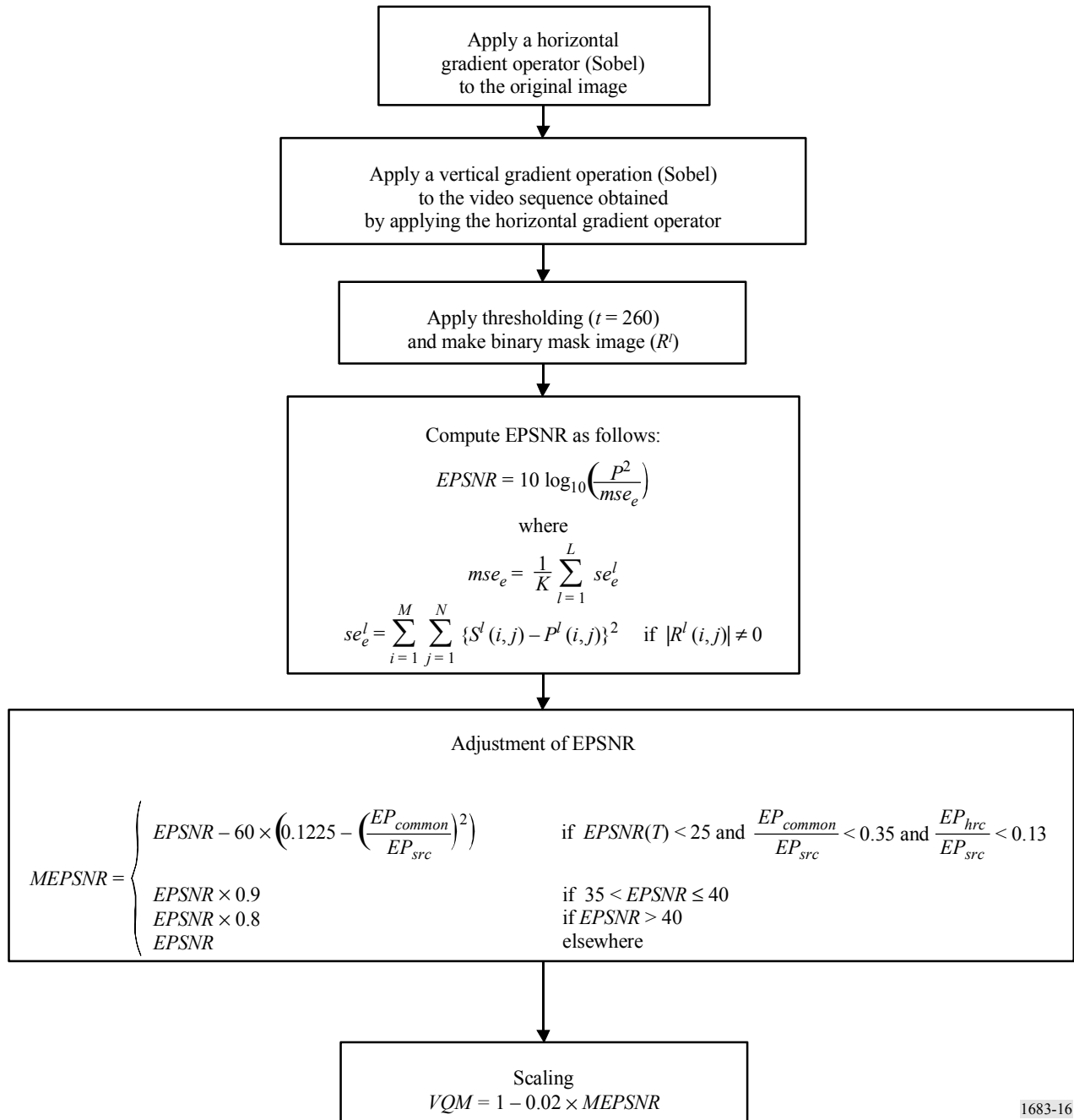
A new model for objective measurement of video quality is proposed based on edge degradation. The model is extremely fast. Once the bit-map is generated, the model is several times faster than the conventional PSNR, providing a significant improvement. Therefore, the model is well suited to applications which require real-time video quality evaluation.

5 Reference

LEE C., EDEN M. and UNSER M., [1998] High quality image resizing using oblique projection operators. *IEEE Trans. Image Processing*, Vol. 5, 5, p. 679-692.

FIGURE 16

The complete block-diagram of the model



1683-16

TABLE 7
The 525 VQM matrix (yonsei_1128c.exe)⁽¹⁾

SRC (image)	HRC																											
	1		2		3		4		5		6		7		8		9		10		11		12		13		14	
1	1	0.679	4	0.525	7	0.512	10	0.419																				
2	2	0.431	5	0.365	8	0.313	11	0.342																				
3	3	0.558	6	0.452	9	0.340	12	0.305																				
4									13	0.668	17	0.581	21	0.556	25	0.535												
5									14	0.543	18	0.485	22	0.443	26	0.410												
6									15	0.631	19	0.477	23	0.441	27	0.411												
7									16	0.467	20	0.415	24	0.376	28	0.346												
8																	29	0.787	35	0.734	41	0.740	47	0.551	53	0.520	59	0.537
9																	30	0.848	36	0.559	42	0.723	48	0.495	54	0.462	60	0.465
10																	31	0.552	37	0.449	43	0.542	49	0.352	55	0.308	61	0.377
11																	32	0.610	38	0.628	44	0.633	50	0.475	56	0.471	62	0.498
12																	33	0.576	39	0.539	45	0.577	51	0.470	57	0.436	63	0.448
13																	34	0.554	40	0.569	46	0.517	52	0.399	58	0.382	64	0.412

⁽¹⁾ After the model was submitted, registration and operator errors were found. The objective data presented in this Annex is the same data as in the VQEG Phase II Final Report. Consequently, when the method described in this Annex is properly implemented, the user may obtain different objective data from those of Table 7.

TABLE 8
The 625 VQM matrix (yonsei_1128c.exe)⁽¹⁾

SRC (image)	HRC																			
	1		2		3		4		5		6		7		8		9		10	
1			4	0.612	10	0.531	16	0.452			29	0.434			42	0.436			52	0.382
2			5	0.544	11	0.540	17	0.451			30	0.437			43	0.440			53	0.363
3			6	0.572	12	0.571	18	0.497			31	0.479			44	0.478			54	0.418
4			7	0.601	13	0.656	19	0.557			32	0.547			45	0.526			55	0.472
5			8	0.603	14	0.621	20	0.500			33	0.492			46	0.444			56	0.390
6			9	0.591	15	0.520	21	0.483			34	0.469			47	0.461			57	0.423
7							22	0.576			35	0.555					48	0.531	58	0.501
8							23	0.512			36	0.500					49	0.482	59	0.457
9							24	0.507			37	0.487					50	0.468	60	0.436
10							25	0.610			38	0.594					51	0.575	61	0.540
11	1	0.753							26	0.594			39	0.508					62	0.485
12	2	0.643							27	0.556			40	0.550					63	0.496
13	3	0.669							28	0.524			41	0.481					64	0.441
⁽¹⁾ After the model was submitted, registration and operator errors were found. The objective data presented in this Annex is the same data as in the VQEG Phase II Final Report. Consequently, when the method described in this Annex is properly implemented, the user may obtain different objective data from those of Table 8.																				

Annex 4**Model 3****CONTENTS**

	<i>Page</i>
1 Introduction	33
2 General description of the IES system	34
3 Correction of offset and gain	36
3.1 Temporal offset.....	36
3.2 Spatial offset	37
3.3 Gain.....	37
4 Image segmentation.....	38
4.1 Plane regions.....	38
4.2 Edge regions	38
4.3 Texture regions	40
5 Objective measurement	40
6 Database of impairment models	40
7 Estimation of impairment models	41
7.1 Computation of W_i	41
7.2 Computation of F_i and G_i	42
8 References	43
Annex 4a	44

1 Introduction

This Annex presents a methodology for video quality assessment using objective parameters based on image segmentation. Natural scenes are segmented into plane, edge and texture regions, and a set of objective parameters are assigned to each of these contexts. A perceptual-based model that predicts subjective (Recommendation ITU-R BT.500 and Recommendation ITU-R BT.802 – Test pictures and sequences for subjective assessments of digital codecs conveying signals produced according to Recommendation ITU-R BT.601) ratings is defined by computing the relationship between objective measures and results of subjective assessment tests, applied to a set of natural