ATIS-0100801.03.2003(R2013)

Digital Transport of One-Way Video Signals – Parameters for Objective Performance Assessment

AMERICAN NATIONAL STANDARD FOR TELECOMMUNICATIONS

As a leading technology and solutions development organization, ATIS brings together the top global ICT companies to advance the industry's most-pressing business priorities. Through ATIS committees and forums, nearly 200 companies address cloud services, device solutions, emergency services, M2M communications, cyber security, ehealth, network evolution, quality of service, billing support, operations, and more. These priorities follow a fast-track development lifecycle – from design and innovation through solutions that include standards, specifications, requirements, business use cases, software toolkits, and interoperability testing.

ATIS is accredited by the American National Standards Institute (ANSI). ATIS is the North American Organizational Partner for the 3rd Generation Partnership Project (3GPP), a founding Partner of oneM2M, a member and major U.S. contributor to the International Telecommunication Union (ITU) Radio and Telecommunications sectors, and a member of the Inter-American Telecommunication Commission (CITEL). For more information, visit < www.atis.org >.

AMERICAN NATIONAL STANDARD

Approval of an American National Standard requires review by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made towards their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

Notice of Disclaimer & Limitation of Liability

The information provided in this document is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE, GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OFMERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. ATIS SHALL NOT BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY ATIS FOR THIS DOCUMENT, AND IN NO EVENT SHALL ATIS BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. ATIS EXPRESSLY ADVISES THAT ANY AND ALL USE OF OR RELIANCE UPON THE INFORMATION PROVIDED IN THIS DOCUMENT IS AT THE RISK OF THE USER.

NOTE - The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights. By publication of this standard, no position is taken with respect to whether use of an invention covered by patent rights will be required, and if any such use is required no position is taken regarding the validity of this claim or any patent rights in connection therewith. Please refer to [http://www.atis.org/legal/patentinfo.asp] to determine if any statement has been filed by a patent holder indicating a willingness to grant a license either without compensation or on reasonable and non-discriminatory terms and conditions to applicants desiring to obtain a license.

ATIS-0100801.03.2003(R2013), *Digital Transport of One-Way Signals – Parameters for Object Performance Assessment*

Is an American National Standard developed by the **ATIS Network Performance, Reliability and Quality of Service Committee (PRQC)**.

Published by

Alliance for Telecommunications Industry Solutions
1200 G Street, NW, Suite 500
Washington, DC 20005

T1.801.03-2003 (R2013)

(Revision of T1.801.03-1996)

American National Standard for Telecommunications -

Digital Transport of One-Way Video Signals – Parameters for Objective Performance Assessment

Secretariat

Alliance for Telecommunications Industry Solutions

Approved September 10, 2003

American National Standards Institute, Inc.

Abstract

This standard provides a video performance estimation method for one-way compressed video signals transported digitally on an error-free network or storage system. This video performance estimation method is for possible use with end-user systems, carriers, information and enhanced-service providers, and customer premise equipment.

Foreword

The information contained in this Foreword is not part of this American National Standard (ANS) and has not been processed in accordance with ANSI's requirements for an ANS. As such, this Foreword may contain material that has not been subjected to public review or a consensus process. In addition, it does not contain requirements necessary for conformance to the standard.

This standard is a revision of T1.801.03-1996. The quality parameters that were used to measure spatial and temporal impairments in the original 1996 standard have been improved to more closely track perceptual changes in quality. In addition, an estimator of overall video quality has been included in this revision that is a linear combination of multiple video quality parameters, each of which measure a different perceptual aspect of quality. These improvements represent 5 years of evolutionary changes to the measurement methods found in the original 1996 standard. The estimator of overall video quality contained in this revision has recently been validated by the Video Quality Experts Group (VQEG) in their Phase II video quality tests [25].

Over the past ten years, the transmission of video using digital compression methods has progressed from limited video conferencing applications to widespread use in applications from high definition television to personal desktop computer communications. Traditional objective video quality measurements designed for analog video systems have been found to correlate poorly with subjective (viewer panel) assessments of these new digital video systems. This is because the perceived quality of digital video systems is generally variable and depends upon dynamic characteristics of both the input video and the digital transmission system. There have been continuing efforts by laboratories and standards organizations to develop new objective measurement methods that accurately track perceived picture quality of digital video systems. In the mid 1990's a series of four standards (T1.801.01-1995 (R2001), T1.801.02-1996 (R2001), T1.801.03-1996, T1.801.04-1997 (R2002)) were issued by Committee T1 that provided background information and a list of parametric calculations to be used in video performance assessment. The parametric calculations contained in T1.801.03-1996 provided the basis for further research and progress in the area of objective video quality measurements.

To address the validation and comparison of video-quality metrics (VQMs), VQEG was formed in 1997 as an informal subgroup of the ITU-T and ITU-R. Over the succeeding two-year period, VQEG designed and implemented extensive subjective and objective test plans to evaluate a number of proponent algorithms. Results from these phase-1 tests were inconclusive in that no single objective model statistically outperformed the others or Peak-Signal-to-Noise-Ratio (PSNR). Hence, no objective methods were recommended to the ITU. In 2001, Committee T1 published a series of five technical reports (T1.TR.72-2001, T1.TR.73-2001, T1.TR.74-2001, T1.TR.75, and T1.TR.77-2002) that documented two of the objective models, their associated normalization procedures (i.e., calibration), and performance attributes.

Many proponents made improvements to their original algorithms and VQEG designed and executed a second series of tests during 2001-2003 for the validation of these improved methods. The results from these phase-2 tests showed that substantial improvements had been made and that one VQM in particular performed extremely well on both 525-line and 625-line formats. That VQM, called the General Model, is the subject of this standard.

This ANSI standard provides a complete description of the General Model and its associated calibration techniques. The methods documented herein are the culmination of 5 years of improvements to the original reduced-reference methods that are described in the prior version of this standard (T1.801.03-1996). Reduced reference parameters utilize features extracted from spatial-temporal regions of the video sequence. In comparison to the uncompressed video stream, these extracted features can be communicated in real time over relatively low bandwidth channels. Hence, these methods can be used to perform in-service video quality monitoring in situations where an ancillary data channel is available to transmit the extracted features between the source and destination ends of the video transmission system under test.

The General Model was designed to be a general purpose VQM for video systems that span a very wide range of quality and bit rates. Extensive subjective and objective tests were conducted to verify the performance of the General Model before it was submitted to VQEG for final validation testing. While the VQEG phase-2 tests only evaluated the performance of the General Model on MPEG-2 and H.263 video systems, the General Model should work well for many other types of coding and transmission systems. The General Model, and its associated automatic calibration techniques (e.g., estimation and correction of spatial registration, temporal registration, and gain/offset errors) have been completely implemented in user-friendly software. This software is available to all interested parties via a no-cost evaluation license agreement (see www.its.bldrdoc.gov/n3/video/vqmsoftware.htm for more information).

Suggestions for improving this standard are welcome and should be sent to the Alliance for Telecommunications Industry Solutions - Committee T1 Secretariat, 1200 G Street N.W., Suite 500, Washington, D.C. 20005.

This standard was processed and approved for submittal to ANSI by Accredited Standards Committee on Telecommunications, T1. Committee approval of the standard does not necessarily imply that all members voted for its approval. At the time it approved this standard, the T1 Committee had the following members:

E.R. Hapeman, T1 Chair

W.R. Zeuch, T1 Vice-Chair

J.A. Crandall, T1 Director

S.M. Carioti, T1 Disciplines

S.D. Barclay, T1 Secretary

C.A. Underkoffler, T1 Chief Editor

S. Wolf, T1A1 Technical Editor

EXCHANGE CARRIERS

Organization Represented	Name of Representative
AT&T Wireless Services, Inc.	Peter Musgrove Brian Daly (Alt.)
BellSouth Telecommunications Inc.	David M. Brady
Qwest	James L. Eitel Richard Prince (Alt.)
Rogers Wireless Inc.	Edward O'Leary Peter Oldfield (Alt.)
SBC Communications, Inc.	Chuck Bailey Bob Hall (Alt.)
Sprint – Local Telecom. Division	Irvin Youngberg
Verizon Communications	Josephine Gallagher Wendy Pugh (Alt.)

GENERAL INTEREST

Organization Represented	Name of Representative
CSI Telecommunications	Michael S. Newman Thomas G. Croda (Alt.)
Cingular Wireless LLC	Don Zelmer Marc Grant (Alt.)
Defense Information Systems Agency	Chris Fitzgerald Don Choi (Alt.)
FBI CALEA Implementation Sect.	Les Szwajkowksi Eric Mason (Alt.)
Microcell Connexions	Venkatesh Sampath Besma Smida (Alt.)
National Communications System	Nicholas Andre F. McClelland (Alt.)
NTIA	Neal B. Seitz
Rural Utilities Service	Orren E. Cameron III Gerald F. Nugent, Jr. (Alt.)
Stanford University	John Cioffi Patricia Oshiro (Alt.)
T-Mobile	Gary K. Jones Mark Younge (Alt.)
Telcordia Technologies	Rick Harrison Cliff Halevi (Alt.)

INTEREXCHANGE CARRIERS

Organization Represented	Name of Representative
AT&T	Charles A. Dvorak Percy Tarapore (Alt.)
Bell Canada	P. Norman Smith
Cable & Wireless	Christophe Liljenstolpe
Intelsat	Mark T. Neibert
Sprint - Long Distance Division	Mark L. Jones
WorldCom	J. Martin Carroll Robert Schafer (Alt.)

MANUFACTURERS

Organization Represented	Name of Representative
Alcatel USA Inc.	Ken Biholar Bill Powell (Alt.)
Beatnik, Inc.	Chris Grigg Chris Muir (Alt.)
Broadcom Corporation	Miguel Peeters Aidan O'Rourke (Alt.)
Catena Networks Inc.	Andrew Deczky Andy Weirich (Alt.)
Centillium Communications, Inc.	Les Brown Guozhu Long (Alt.)
Cisco Systems, Inc.	John McDonough John Krahner (Alt.)
Ericsson Inc.	Asok Chatterjee
Excelsus Technologies Inc.	William J. Buckley Don Robert House (Alt.)
Flarion Technologies Inc.	Mark Klerer
Fujitsu America Inc.	Mike White
GlobespanVirata, Inc.	Massimo Sorbara Clete Gardenhour (Alt.)
Harris Corp.	Marlis Humphrey
Hatteras Networks	Manu Kaycee Matt Squire (Alt.)
Hewlett-Packard	Steve Mills Karen Higginbottom (Alt.)

Organization Represented	Name of Representative
Ikanos Communications	Ed Eckert Sam Heidari (Alt.)
Juniper Networks, Inc.	Elizabeth Lytle
Lucent Technologies	Rick Townsend Wayne R. Zeuch (Alt.)
Mangrove Systems, Inc.	Betsy Gilbert Jonathan Reeves (Alt.)
Mindspeed Technologies, Inc.	Keith Chu Trey Malpass (Alt.)
Motorola Inc.	Brye Bonner Bernard Dugerdil (Alt.)
Next Level Communications	Sabit Say Jeff Weber (Alt.)
Nokia Telecommunications Inc.	Chris Wallace Ed Ehrlich (Alt.)
Nortel Networks	Subhash Patel Joseph A. Zebarth (Alt.)
Qualcomm Inc.	Mark Epstein Ed Tiedemann (Alt.)

Organization Represented	Name of Representative
Siemens Information & Communications Networks, Inc.	David E. Francisco
Skyworks Solutions Inc.	Arun Arunachalam Naren Jauhal (Alt.)
STMicroelectronics	Sabina Fanfoni Srikanth Gopalan (Alt.)
Symmetricom Inc.	Don Skipwith Phil Mann (Alt.)
Tellabs Operations, Inc.	William A. Walker Rick Younce (Alt.)
Tellium, Inc.	Krishna Bala, PhD Siegfried Giebl (Alt.)
Texas Instruments	Krista Jacobsen Thomas Maudoux (Alt.)
TranSwitch Corp.	Jitender Vij Edwin Soltysiak (Alt.)
TruePosition Inc.	Thomas Ginter Rhys Robinson (Alt.)
Westell Technologies, Inc.	Bruce Kuhn Tim Duitsman (Alt.)

At the time it approved this standard, Technical Subcommittee T1A1 on Performance, Reliability, and Signal Processing, which is responsible for the development of this standard, had the following members:

R. Wohlert, T1A1 Chair
N. Seitz, T1A1 Vice-Chair

Organization Represented	Name of Representative
Alcatel USA Inc.	Ken Biholar
AT&T	Percy Tarapore Alfred Morton (Alt.)
BellSouth Telecommunications Inc.	Eric Hauch Archie McCain (Alt.)
CSI Telecommunications	Michael S. Newman Thomas G. Croda (Alt.)
Ericsson Incorporated	Mustafa Kocaturk Sangamesh Vinayagamurthy (
Intelsat	Mart T. Neibert
Lucent Technologies	Stuart O. Goldman
National Communications System	An Nguyen i. Furey (Alt.)
NTIA	Neal B. Seitz

Organization Represented	Name of Representative
Nortel Networks	Subhash Patel Oscar Avellaneda (Alt.)
Qwest	Bill Wycoff David Clark (Alt.)
SBC Communications, Inc.	Randolph Wohlert Pierre Costa (Alt.)
Siemens Information and Communication Networks, Inc.	Suhas. S. Gandhi David E. Francisco (Alt.)
Sprint – Long Distance Division	Mark L. Jones
Telcordia Technologies	Spilios Makris
Verizon Communications	Wendy Pugh

Working Group T1A1.3 on Performance of Digital Networks and Services, which was responsible for the development of this standard, had the following members:

Greg Cermak	Verizon
John Colombo	Verizon
Suhas S. Gandhi	Siemens Carrier Networks
William HB Greer	Bell South Telecommunications
Pierre Costa	SBC
Chuck Dvorak	AT&T
David Fibush	Tektronix
Keith Mainwaring	Cisco Systems
Alfred Morton	AT&T
Mark Neibert	INTELSAT
Neal B. Seitz	NTIA/ITS
Peter Shelus	Telcordia Technologies
Arthur Webster	NTIA/ITS.T
Randolph Wohlert	SBC
Stephen Wolf	NTIA/ITS.T
W. R. Wycoff	Qwest Corporation
Joseph Zebarth	Nortel Networks

Table of Contents

Table of Figures

Table of Tables

American National Standard for Telecommunications –

Digital Transport of One-Way Video Signals – Parameters for Objective Performance Assessment

1 Scope, Purpose, and Application

1.1 Scope

This standard specifies a method for estimating the video performance of a one-way video transmission system, as shown in Figure 1. The objective video performance estimator is defined for the end-to-end transmission quality between the two points shown in the figure. The estimation method is based on processing 8-bit digital component video as defined by ITU-R Recommendation BT.601 (henceforth abbreviated as Rec. 601).[1] The encoder shown in Figure 1 can utilize various compression methods (e.g., MPEG, H.263, NTSC, etc.). The transmission channel may be a simple pass-through for evaluation of a codec (encoder/decoder combination) or may include a concatenation of various compression methods and memory storage devices, but it is assumed that any digital transport involved is error-free. While the derivation of the objective quality estimator described in this standard considered error impairments (e.g., bit errors, dropped packets), independent testing results are not currently available to support the use of the estimator for systems with error impairments.

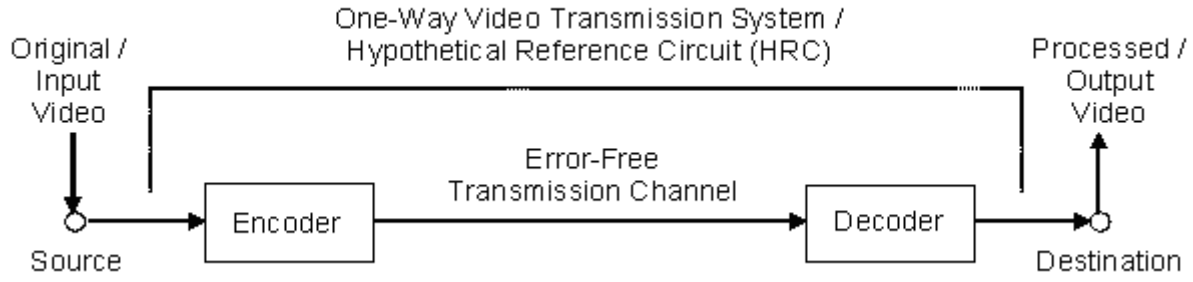

Figure 1 - Reference model for measurement of one-way video transmission system performance

1.2 Purpose

This standard provides a video performance estimation method for one-way compressed video signals transported digitally on an error-free network or storage system. This video performance estimation method is for possible use with end-user systems, carriers, information and enhanced-service providers, and customer premise equipment, provided they introduce no significant error events (e.g., bit errors, dropped packets).

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

1.3 Application

This standard provides 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 General Model described in this standard include but are not limited to:

1. Codec evaluation, specification, and acceptance testing, consistent with the limited accuracy as described below.
2. Real-time, in-service quality monitoring at the source.
3. Remote destination quality monitoring when a copy of the source is available.
4. Real-time, in-service quality monitoring at the source or destination when an ancillary data channel is available to transmit the extracted feature information (see clauses 4 and 7.1.1).
5. 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).

Alternate standardized quality estimation models may satisfy these applications with similar accuracy. Users of this standard may consult the comparable specifications of the ITU-T and ITU-R.

1.3.1 Limitations

The General Model described in this standard cannot be used to replace subjective testing. Correlations 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 standard does not supply a means for quantifying potential errors between subjective assessments of quality and the General Model's estimates. Users of this standard should review the comparison of available subjective and objective results to gain an understanding of the range of video quality rating estimation error.

The General Model should yield acceptable results for multimedia video class MM5 (defined in ITU-T Recommendation P.911, Annex B). However, use of General Model for MM5 video systems is not recommended until further independent validation tests can be performed.

Use of the General Model for video systems with transmission channel errors (e.g., packet loss; see Figure 1) is also not recommended until further independent validation tests can be performed.

The General Model does not provide a comprehensive evaluation of two-way multimedia transmission quality. It only measures the effects of one-way video distortion. The effects of audio distortion, audio delay, video delay, audio-video synchronization, and other impairments related to two-way interaction are not reflected in the General Model scores. Therefore, it is possible to have high General Model scores, yet poor quality of the multimedia connection overall.

2 Normative References

– ITU-R Recommendation BT.601 (10/95), *Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-Screen 16:9 Aspect Ratios.* [2]

[2] This document is available from the International Telecommunications Union, Radiocommunication Sector.
< http://www.itu.int/ITU-R/ >

– ITU-T Recommendation P.911 (12/1998), *Subjective Audiovisual Quality Assessment Methods for Multimedia Applications.* [3]

3 Definitions, Abbreviations, and Acronyms

3.1 Definitions

3.1.1 4:2:2: A Y, C_B, C_R image sampling format where chrominance planes (C_B and C_R) are sampled horizontally at half the luminance (Y) plane's sampling rate. See Rec. 601 (clause 2).

3.1.2 Absolute Temporal Information (ATI): A feature derived from the absolute value of temporal information images that are computed as the difference between successive frames in a video clip. ATI quantifies the amount of motion in a video scene. See clause 7.5 for the precise mathematical definition.

3.1.3 American National Standards Institute (ANSI): Administrator and coordinator of the United States private sector voluntary standardization system.

3.1.4 Alliance for Telecommunications Industry Solutions (ATIS): A North American standards body that develops telecommunications standards, operating procedures, and guidelines through its sponsored committees and forums.

3.1.5 Big YUV: The binary file format used for storing clips that have been sampled according to Rec. 601. In the Big YUV format, all the video frames for a scene are stored in one large binary file, where each individual frame conforms to Rec. 601 sampling. The Y represents the luminance channel information, the U represents the blue color difference channel (i.e., C_B in Rec. 601), and the V represents the red color difference channel (i.e., C_R in Rec. 601). The pixel ordering in the binary file is the same as that specified in SMPTE 125M [17]. The full specification of the Big YUV file format is given in clause 5 and software routines for reading and displaying Big YUV files are given in [24].

3.1.6 Clip: Digital representation of a scene that is stored on computer media.

3.1.7 Clip VQM: The VQM of a single clip of processed video.

3.1.8 Chrominance (C, C_B, C_R): The portion of the video signal that predominantly carries the color information (C), perhaps separated further into a blue color difference signal (C_B) and a red color difference signal (C_R).

3.1.9 Codec: Abbreviation for a coder/decoder or compressor/decompressor.

3.1.10 Common Intermediate Format (CIF): A video sampling structure used for video teleconferencing where the luminance channel is sampled at 352 pixels by 288 lines [11].

3.1.11 Feature: A quantity of information associated with, or extracted from, a spatial-temporal sub-region of a video stream (either an original video stream or a processed video stream).

3.1.12 Field: One half of a frame, containing all of the odd or even lines.

3.1.13 Frame: One complete television picture.

3.1.14 Frames per Second (FPS): The number of original frames per second transmitted by the video system under test. For instance, an NTSC video system transmits approximately 30 FPS.

3.1.15 Gain: A multiplicative scaling factor applied by the hypothetical reference circuit (HRC) to all pixels of an individual image plane (e.g., luminance, chrominance). Gain of the luminance signal is commonly known as *contrast*.

[3] This document is available from the International Telecommunications Union, Telecommunication Standardization Sector < http://www.itu.int/ITU-T/ >.

3.1.16 General Model: The video quality model, that is the subject of this standard (clause 9). The General Model was submitted to the phase-2 tests performed by the Video Quality Experts Group (VQEG). The VQEG Phase-2 final report describes the performance of the General Model (see [25], proponent H).

3.1.17 H.261: Abbreviation for ITU-T Recommendation H.261 [11].

3.1.18 Hypothetical Reference Circuit (HRC): A video system under test such as a codec or digital video transmission system.

3.1.19 Input Video: Video before being processed or distorted by an HRC (see Figure 2). Input video may also be referred to as *Original Video*.

3.1.20 Institute for Radio Engineers (IRE) Unit: A unit of voltage commonly used for measuring video signals. One IRE is equivalent to 1/140 of a volt.

3.1.21 International Telecommunication Union (ITU): An international organization within the United Nations System where governments and the private sector coordinate global telecommunications networks and services. The ITU includes the Radiocommunication Sector (ITU-R) and the Telecommunication Standardization Sector (ITU-T).

3.1.22 Luminance (Y): The portion of the video signal that predominantly carries the luminance information (i.e., the black and white part of the picture).

3.1.23 Mean Opinion Score (MOS): The average subjective quality judgment assigned by a panel of viewers to a processed video clip.

3.1.24 Moving Picture Experts Group (MPEG): A working group of ISO/IEC in charge of the development of standards for coded representation of digital audio and video (e.g., MPEG-1, MPEG-2, MPEG-4).

3.1.25 National Television Systems Committee (NTSC): The 525-line analog color video composite system [18].

3.1.26 Offset or level offset: An additive factor applied by the HRC to all pixels of an individual image plane (e.g., luminance, chrominance). Offset of the luminance signal is commonly known as *brightness*.

3.1.27 Original Region of Interest (OROI): A Region of Interest (ROI) extracted from the original video, specified in Rectangle Coordinates.

3.1.28 Original Video: Video before being processed or distorted by an HRC (see Figure 2). Original video may also be referred to as input video since this is the *video input* to the digital video transmission system.

3.1.29 Original Valid Region (OVR): The Valid Region of an original video clip, specified in Rectangle Coordinates.

3.1.30 Output Video: Video that has been processed or distorted by an HRC (see Figure 2). Output video may also be referred to as *Processed Video*.

3.1.31 Over-scan: The portion of the video that is not normally visible on a standard television monitor.

3.1.32 Phase-Altering Line (PAL): The 625-line analog color video composite system.

3.1.33 Parameter: A measure of video distortion that is the result of comparing two parallel streams of features, one stream from the original video and the corresponding stream from the processed video.

3.1.34 Processed Region of Interest (PROI): A Region of Interest (ROI) extracted from the processed video and corrected for spatial shifts of the HRC, specified in Rectangle Coordinates.

3.1.35 Processed Video: Video that has been processed or distorted by an HRC (see Figure 2). Processed video may also be referred to as *output video* since this is the video output from the digital video transmission system.

3.1.36 Processed Valid Region (PVR): The Valid Region of a processed video clip from an HRC, specified in Rectangle Coordinates. The PVR is always referenced to the original video, so it is necessary to correct for any spatial shifts of the video by the HRC before computing PVR. Thus, PVR is always contained within the OVR. The region between the PVR and the OVR is that portion of the video that was blanked or corrupted by the HRC.

3.1.37 Production Aperture: The image lattice that represents the maximum possible image extent in a given standard. The Production Aperture represents the desirable extent for image acquisition, generation, and processing, prior to blanking. For Rec. 601 sampled video, the Production Aperture is 720 pixels x 486 lines for 525-line systems and 720 pixels x 576 lines for 625-line systems [19].

3.1.38 Quarter Common Intermediate Format (QCIF): A video sampling structure used for video teleconferencing where the luminance channel is sampled at 176 pixels by 144 lines [11].

3.1.39 Rec. 601: Abbreviation for ITU-R Recommendation BT.601 (clause 2), a common 8-bit video sampling standard that samples the luminance (Y) channel at 13.5 MHz, and the blue and red color difference channels (C_B and C_R) at 6.75 MHz. See clause 5 for more information.

3.1.40 Rectangle Coordinates: A rectangular shaped image sub-region that is completely contained within the production aperture and that is specified by four coordinates (top, left, bottom, right). Numbering starts from zero so that the (top, left) corner of the sampled image is (0, 0). See clause 5.3.

3.1.41 Reduced-Reference: A video quality measurement methodology that utilizes low bandwidth features extracted from the original or processed video streams, as opposed to using full-reference video that requires complete knowledge of the original and processed video streams [12]. Reduced-reference methodologies have advantages for end-to-end in-service quality monitoring since the reduced-reference information is easily transmitted over ubiquitous telecommunications networks.

3.1.42 Reframing: The process of reordering two consecutively sampled interlaced fields of processed video into a frame of video. Reframing is necessary when HRCs do not preserve standard interlace field types (e.g., an NTSC field type one is output as an NTSC field type two and vice versa). See clause 6.1.2.

3.1.43 Region of Interest (ROI): An image lattice (specified in Rectangle Coordinates) that is used to denote a particular sub-region of a field or frame of video. Also see SROI.

3.1.44 Scene: A sequence of video frames.

3.1.45 Spatial Information (SI): A feature based on statistics that are extracted from the spatial gradients (i.e., edges) of an image or video scene. References [3] and [13] provide a definition of SI based on statistics extracted from 3 x 3 Sobel-filtered images [16] while clause 7.2.2 of this standard provides a definition of SI based on statistics extracted from much larger 13 x 13 edge-filtered images (Figure 12).

3.1.46 Spatial Region of Interest (SROI): The specific image lattice (specified in Rectangle Coordinates) that is used to calculate the VQM of a video clip. The SROI is a rectangular subset that lies completely inside the Processed Valid Region. For Rec. 601 sampled video, the recommended SROI is 672 pixels x 448 lines for 525-line systems and 672 pixels x 544 lines for 625-line systems, centered within the Production Aperture. This recommended SROI corresponds to approximately the portion of the video picture that is visible on a monitor, excluding the over-scan area. Also see *ROI*.

3.1.47 Spatial Registration: The process that is used to estimate and correct for spatial shifts of the processed video sequence with respect to the original video sequence.

3.1.48 Spatial-Temporal (S-T) Sub-Region: A block of image pixels in an original or processed video stream that includes a vertical extent (number of rows), a horizontal extent (number of columns), and a time extent (number of frames). See Figure 10.

3.1.49 Society of Motion Picture and Television Engineers (SMPTE): An industry-leading society for the motion picture and television industries devoted to advancing theory and application in motion imaging, including film, television, video, computer imaging, and telecommunications. The industry relies

on SMPTE to generate standards, engineering guidelines, and recommended practices to be followed by respective field professionals.

3.1.50 Temporal Information (TI): A feature based on statistics that are extracted from the temporal gradients (i.e., motion) of a video scene. References [3], [13] and clause 7.5 of this standard all provide definitions of TI based on statistics extracted from simple frame differences.

3.1.51 Temporal Region of Interest (TROI): The specific time segment, sequence, or subset of frames that is used to calculate a clip's VQM. The TROI is a contiguous segment of frames that lies completely inside the Temporal Valid Region. The maximum possible TROI is the fully registered time segment and contains all temporally registered frames within the TVR. If reframing is required, the processed clip is always reframed, not the original clip.

3.1.52 Temporal Registration: The process that is used to estimate and correct for the temporal shift (i.e., video delay) of the processed video sequence with respect to the original video sequence (see clause 6.4.1).

3.1.53 Temporal Valid Region (TVR): The maximum time segment, sequence, or subset of video frames that may be used for calibration and VQM calculation. Frames outside of this time segment will always be considered invalid.

3.1.54 Uncertainty (U): The estimated error (plus or minus) in the temporal registration after allowance is made for the best guess of the HRC video delay. See clause 6.4. (The abbreviation "U" may also be used to denote blue color difference channel; see *Big YUV*.)

3.1.55 Valid Region (VR): The rectangular portion of an image lattice (specified in Rectangle Coordinates) that is not blanked or corrupted due to processing. The Valid Region is a subset of the production aperture of the video standard and includes only those image pixels that contain picture information that has not been blanked or corrupted. See *Original Valid Region* and *Processed Valid Region*.

3.1.56 Video Quality Experts Group (VQEG): A group of international video quality experts that conduct validation tests for objective video performance metrics. Results from VQEG are forwarded to the ITU and may be used as the basis for international video quality measurement recommendations.

3.1.57 Video Quality Metric, Model, or Measurement (VQM): An overall measure of video impairment (see *Clip VQM*, *General Model*). VQM is reported as a single number and has a nominal output range from zero to one, where zero is no perceived impairment and one is maximum perceived impairment.

3.2 Abbreviations & Acronyms

ANSI	American National Standards Institute
ATI	Absolute Temporal Information
ATIS	Alliance for Telecommunications Industry Solutions
C	Combined Chrominance Signal
C_B	Blue Color Difference Signal (see also U)
CIF	Common Intermediate Format
C_R	Red Color Difference Signal (see also V)
FPS	Frames per Second
HRC	Hypothetical Reference Circuit
IEC	International Electrotechnical Commission *or* Incoming Error Count
IRE	Institute for Radio Engineers Unit
ISO	International Organization for Standardization
ITU-R	International Telecommunications Union – Radiotelecommunications Sector

ITU-T	International Telecommunications Union – Telecomunication Standardization Sector
MOS	Mean Opinion Score
MPEG	Moving Pictures Expert's Group
NTSC	National Television System Committee
OROI	Original Region of Interest
OVR	Original Valid Region
PAL	Phase-Altering Line
PROI	Processed Region of Interest
PSNR	Peak-Signal-to-Noise-Ratio
PVR	Processed Valid Region
QCIF	Quarter Common Intermediate Format
ROI	Region of Interest
SI	Spatial Information
SMPTE	Society of Motion Picture and Television Engineers
SROI	Spatial Region of Interest
S-T	Spatial-Temporal
TI	Temporal Information
TROI	Temporal Region of Interest
TVR	Temporal Valid Region
U	Blue Color Difference Signal (see also C_B) *or* Uncertainty
V	Red Color Difference Signal (see also C_R)
VQEG	Video Quality Experts Group
VQM_G	General Video Quality Metric
VR	Valid Region
Y	Luminance
YUV	Denotes 4:2:2 or Rec. 601 sampled video files

4 Overview of the Video Quality Metric (VQM) Computation

This standard provides a complete description of the General Model and its associated calibration algorithms. These automated objective measurement algorithms provide close approximations to the overall quality impressions, or *mean opinion scores*, of digital video impairments that have been graded by panels of viewers. Figure 2 gives an overview diagram of the processes required to compute the General VQM. These processes include sampling of the original and processed video streams (clause 5), calibration of the original and processed video streams (clause 6), extraction of perception-based features (clause 7), computation of video quality parameters (clause 8), and calculation of the General Model (clause 9). The General Model tracks the perceptual changes in quality due to distortions in any component of the digital video transmission system (e.g., encoder, digital channel, decoder).

The method of measurement documented herein utilizes high bandwidth reduced-reference parameters [12]. These reduced reference parameters utilize features extracted from spatial-temporal (S-T) regions of the video sequence (see clause 7.1.1). Hence, the method of measurement presented here may also be used to perform in-service video quality monitoring in situations where an ancillary data channel is available to transmit the extracted features between the source and destination ends of an HRC as shown in Figure 2.

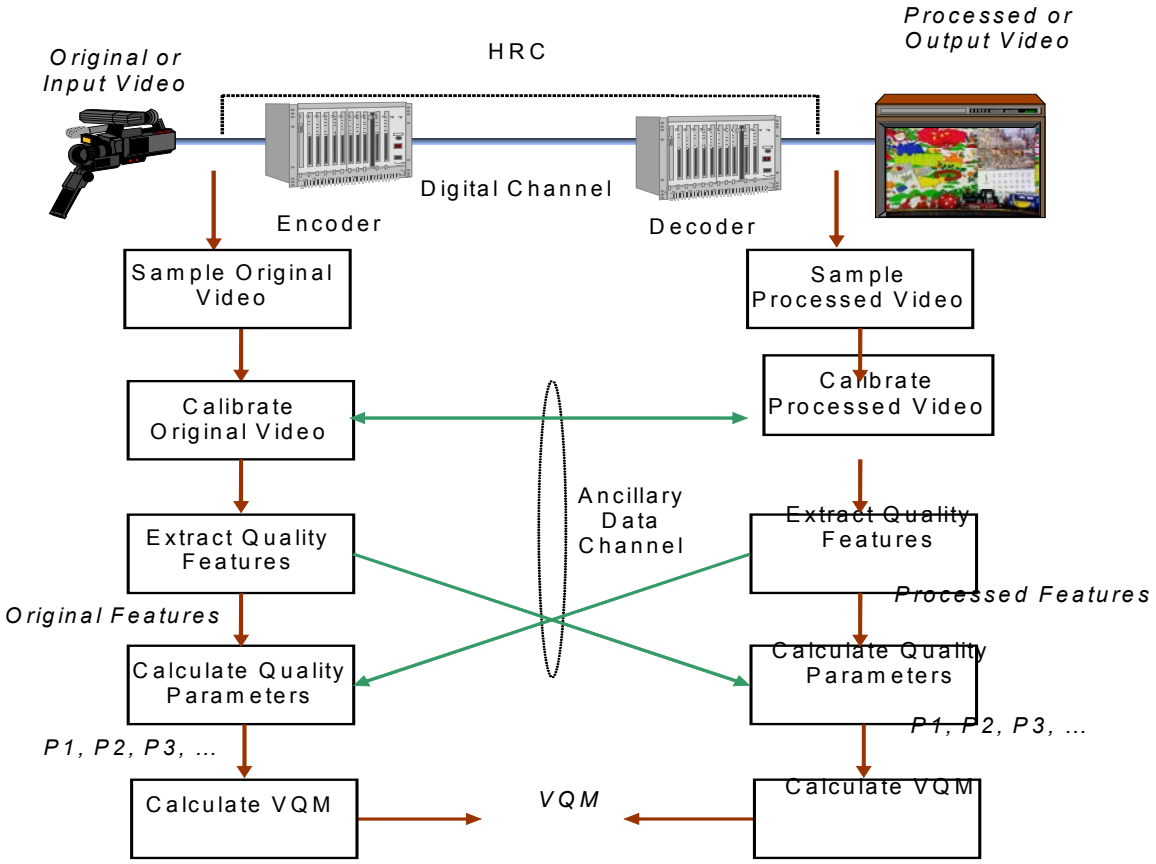

Figure 2 - Steps required to compute VQM

5 Sampling

The computer-based algorithms in this standard assume that the original and processed video streams are available as digital representations stored on computer media (referred to as a clip in this standard). If the video is analog format, one of the most widely used digital sampling standards is Rec. 601 (clause 2). Composite video such as NTSC and PAL must first be converted into component video that contains the following three signals: luminance (Y), blue color difference (C_B), and red color difference (C_R). Rec. 601 sampling is also commonly known as 4:2:2 sampling since the Y channel is sampled at full rate while the C_B and C_R channels are sampled at half rate. Rec. 601 specifies a 13.5 MHz sample rate that produces 720 Y samples per video line. Since there are 486 lines that contain picture information in the 525-line NTSC standard, the complete Rec. 601 sampled Y video frame will be 720 pixels by 486 lines. Likewise, when 625-line PAL video is sampled according to Rec. 601, the Y video frame will contain 720 pixels by 576 lines. If 8 bits are used to uniformly sample the Y signal, Rec. 601 specifies that reference black (i.e., 7.5 IRE units) be sampled as a “16” and reference white (i.e., 100 IRE units) be sampled as a “235.” Thus, a working margin is available for video signals that exceed the reference black and white levels before they are clipped by the analog to digital converter. The chrominance channels (C_B and C_R) are each sampled at 6.75 MHz such that the first pair of chrominance samples (C_B, C_R) is associated with the first Y luminance sample, the second pair of chrominance samples is associated with the third luminance sample, and so forth. Since the chrominance channels are bipolar, zero signal is sampled as a “128.”

5.1 Temporal Indexing of Original and Processed Video Files

A luminance video frame that results from Rec. 601 sampling will be denoted as **Y**(*t*). The variable *t* is being used here as an index for addressing the sampled frames within the original and processed Big YUV files; it does not denote actual time. If the Big YUV file contains N frames, as shown in Figure 2, *t* = 0 denotes the first frame that was sampled and *t* = (N-1) denotes the last frame that was sampled.

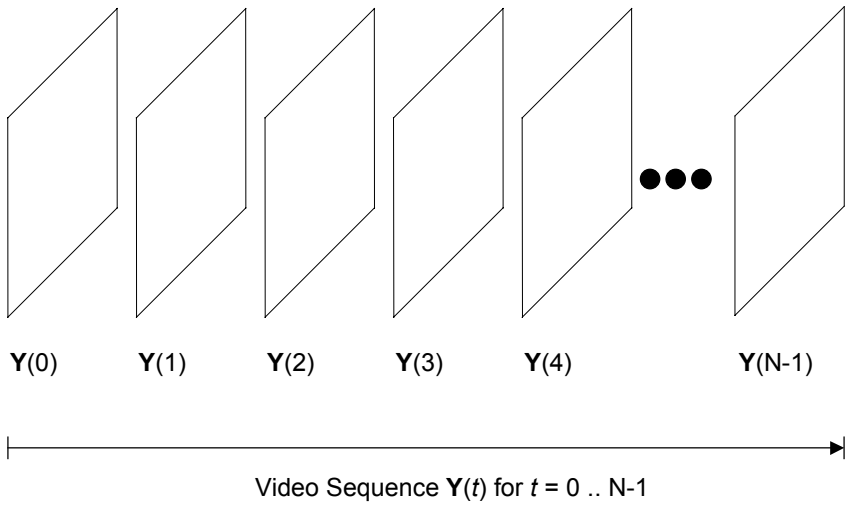

Figure 3 - Temporal indexing of frames in Big YUV files

All the algorithms are written and described from the viewpoint of operation on sampled file pairs: one original video sequence and an associated processed video sequence. To avoid confusion, both files are assumed to be the same length. Furthermore, an initial assumption will be made that the first frame of the original file aligns temporally to the first frame of the processed file, within plus or minus some temporal uncertainty.

For real-time, in-service implementations, this balanced uncertainty presumption can be replaced with a one-sided uncertainty. Causality constrains the range of temporal uncertainty. For example, a processed frame occurring at time *t* = n must come from original frames occurring at or before time *t* = n.

The above assumption regarding original and processed video files (i.e., that the first frames align) is equivalent to selecting the best guess for the temporal delay of the HRC shown in Figure 2. Therefore, the uncertainty that remains in the video delay estimate will be denoted as plus or minus **U**.

5.2 Spatial Indexing of Original and Processed Video Frames

The coordinate system used for the sampled luminance frames is shown in Figure 4. The horizontal and vertical coordinates of the upper left corner of the luminance frames are defined to be (v = 0, h = 0), where the horizontal axis (h) coordinate values increase to the right and the vertical axis (v) coordinate values increase down. Horizontal axis coordinates range from 0 to one less than the number of pixels in a line. Vertical axis coordinates range from 0 to one less than the number of lines in the image, which will be specified in frame lines for progressive systems and either field lines or frame lines for interlace systems. The amplitude of a sampled pixel in **Y**(*t*) at row *i* (i.e., v = *i*), column *j* (i.e., h = *j*), and time *t* is denoted as *Y*(*i*, *j*, *t*).

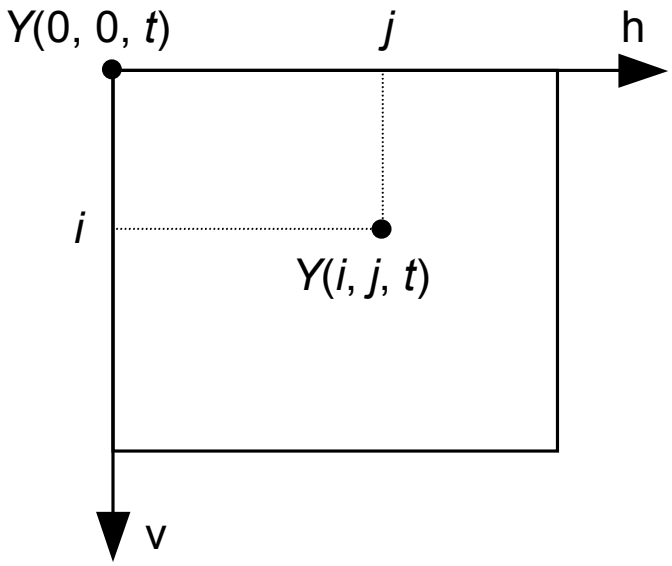

Figure 4 - Coordinate system used for sampled luminance Y frames

A clip of video sampled according to Rec. 601 is stored in "Big YUV" file format, where the Y denotes the Rec. 601 luminance information, the U denotes the blue color-difference information (i.e., C_B in Rec. 601), and the V denotes the red color-difference information (i.e., C_R in Rec. 601). In the Big YUV file format, all the frames are stored sequentially in one large continuous binary file. The image pixels are stored sequentially by video scan line as bytes in the following order: C_{B0}, Y_0, C_{R0}, Y_1, C_{B2}, Y_2, C_{R2}, Y_3, etc., where the numerical subscript denotes the pixel number (pixel replication or interpolation must be used to find the C_B and C_R chrominance samples for Y_1, Y_3, …). This byte ordering is equivalent to that specified in SMPTE 125M [17].

5.3 Specifying Rectangular Sub-Regions

Rectangular sub-regions of a sampled image are used to control the computation of VQM. For instance, VQM may be computed over the valid region of the sampled image or over a user-specified spatial region of interest that is smaller than the valid region. Specification of rectangular sub-regions will use rectangle coordinates defined by the four quantities *top*, *left*, *bottom*, and *right*. Figure 5 illustrates the specification of a rectangular sub-region for a single frame of sampled video. The red image pixels are included in the sub-region but the black image pixels are excluded. In the calculation of VQM, an image is often divided into a large number of smaller sub-regions that abut. The rectangular sub-region definition used in Figure 5 defines the grid used to display these abutted sub-regions and the math used to extract features from each abutted sub-region.

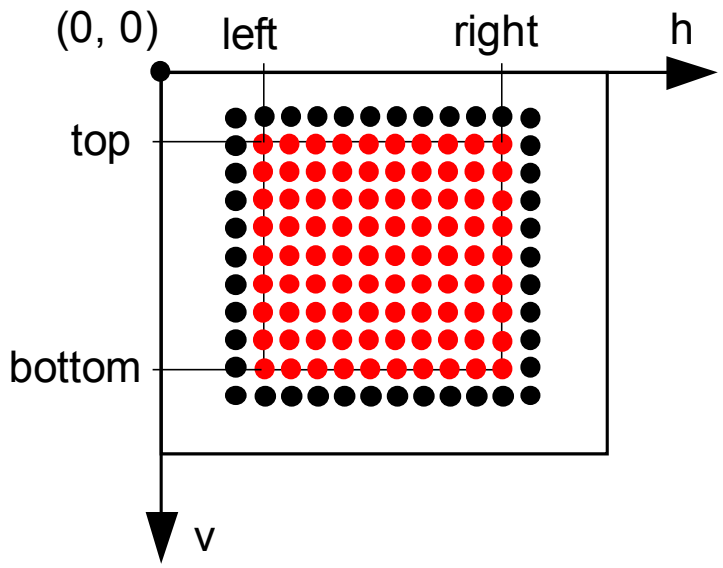

Figure 5 - Rectangle coordinates for specifying image sub-regions

5.4 Considerations for Video Sequences Longer Than 10 Seconds

The video quality measurements in this standard were based upon subjective test results that utilized 8 to 10 second video clips. When working with longer video sequences, the sequence should be divided into shorter video segments, where each segment is assumed to have its own calibration and quality attributes. Dividing the video stream into overlapping segments and processing each segment independently is one method for emulating continuous quality assessments for long video sequences using the VQM techniques presented herein.

6 Calibration

Four steps are required to properly calibrate the sampled video in preparation for feature extraction. These steps are:

1. Spatial registration estimation and correction;
2. Valid region estimation to limit the extraction of features to those pixels that contain picture information;
3. Gain and level offset estimation and correction (commonly known as contrast and brightness), and;
4. Temporal registration estimation and correction.

Step 2 must be performed on both the original and processed video streams. Steps 1, 3, and 4 must be performed on the processed video stream. Normally, the spatial registration, gain, and level offset are constant for a given video system and hence these quantities only need to be calculated once. However, it is common for the valid region and temporal registration to change depending upon scene content. For instance, full screen and letterbox scenes will have different valid regions, and videoconferencing systems often have variable video delays that depend upon scene content (e.g., talking head versus sports action). In addition to the calibration techniques presented here, the reader may also want to examine [6] and [14] (see Annex A) for alternate spatial and temporal registration methods.

Calibrating prior to feature extraction means that VQM will not be sensitive to horizontal and vertical shifts of the image, temporal shifts of the video stream that result from non-zero video delays, and changes in image contrast and brightness that fall within the dynamic range of the video sampling unit. While these calibration quantities can have a significant impact on the overall perceived quality (e.g., low contrast images from a video system with a gain of 0.3), the philosophy taken here is to report calibration information separately from VQM. Spatial shifts, valid regions, gains, and offsets can normally be adjusted using good engineering practices, while temporal delays provide important quality information when evaluating two-way or interactive video systems.

All of the video quality features and parameters (clauses 7 and 8) assume that only one video delay will be removed to temporally register the processed video sequence (i.e., constant video delay). Some video systems or HRCs delay individual processed frames by different amounts (i.e., variable video delay). For the purposes of this standard, all video systems are treated as having a constant video delay. Variations from this delay are considered degradations that are measured by the features and parameters. This approach appears to yield higher correlations to subjective score than video quality measurements based on processed video sequences where variable video delay has been removed. When working with long video sequences (see clause 5.4), the sequence should be divided into shorter video segments, where each segment has its own constant video delay. This allows for some delay variation as a function of time. A more continuous estimation of delay variations may be obtained by dividing the sequence into overlapping time segments.

If the HRC being tested also spatially scales the picture or changes its size (e.g., zoom), then an additional step to estimate and remove this spatial scaling would have to be included in the calibration process. Spatial scaling is beyond the scope of this standard.

6.1 Spatial Registration

6.1.1 Overview

The spatial registration process determines the horizontal and vertical spatial shift of the processed video relative to the original video. A positive horizontal shift is associated with a processed image that has been moved to the right by that number of pixels. A positive vertical shift is associated with a processed image that has been moved down that number of lines. Thus, spatial registration of interlace video results in three numbers:

1. The horizontal shift in pixels;
2. The vertical field one shift in field lines; and
3. The vertical field two shift in field lines.

Spatial registration of progressive video results in two numbers:

1. The horizontal shift; and
2. The vertical shift in frame lines.

The accuracy of the spatial registration algorithm is to the nearest pixel for horizontal shifts and to the nearest line for vertical shifts. After the spatial registration has been calculated, the spatial shift is removed from the processed video stream (e.g., a processed image that was shifted down is shifted back up). For interlace video, this may include reframing of the processed video stream as implied by comparison of the vertical field one and two shifts.

When operating on interlace video, all operations will consider video from each field separately; when operating on progressive video, all operations will consider the entire video frame simultaneously. For simplicity, the spatial registration algorithm will first be entirely described for interlace video, this being the more complicated case. The modifications needed to operate on progressive video are identified in 6.1.6.

Spatial registration must be determined before processed valid region (PVR), gain and level offset, and temporal registration. Specifically, each of those quantities must be computed by comparing original and processed video content that has been spatially registered. If the processed video stream were spatially shifted with respect to the original video stream and this spatial shift were not corrected, then these estimates would be corrupted because they would be based on dissimilar video content. Unfortunately, spatial registration cannot be correctly determined unless the PVR, gain and level offset, and temporal registration are also known. The interdependence of these quantities produces a "chicken or egg" measurement problem. Calculation of the spatial registration for one processed field requires that one know the PVR, gain and level offset, and the closest matching original field. However, one cannot determine these quantities until the spatial shift is found. A full exhaustive search over all variables would require a tremendous number of computations if there were wide uncertainties in the above quantities.

The solution presented here performs an iterative search to find the closest matching original field for each processed field. This search includes iteratively updating estimates for PVR, gain and level offset, and temporal registration. For some processed fields, however, the spatial registration algorithm could fail. Usually, when the spatial registration is incorrectly estimated for a processed field, the ambiguity is due to characteristics of the scene. Consider, for example, a digitally-created interlace scene containing a pan to the left. Because the pan was computer generated, this scene could have a horizontal pan of exactly one pixel every field. From the spatial registration search algorithm's point of view, it would be impossible to differentiate between the correct spatial registration computed using the matching original field, and a two pixel horizontal shift computed using the field that occurs two fields prior to the matching original field. For another example, consider an image consisting entirely of digitally perfect black and white vertical lines. Because the image contains no horizontal lines, the vertical shift is entirely ambiguous. Because the pattern of vertical lines repeats, the horizontal shift is ambiguous, two or more horizontal shifts being equally likely.

Therefore, the iterative search algorithm should be applied to a sequence of processed fields. The individual estimates of spatial shifts from multiple processed fields can then be used to produce a more robust estimate. Spatial shift estimates from multiple sequences or scenes may be further combined to produce an even more robust estimate for the HRC being tested; assuming that the spatial shift is constant for all scenes passing through the HRC.

6.1.2 Interlace Issues

Vertical spatial registration of interlaced video is a greater challenge than progressive video, since the spatial registration process must differentiate between field one and field two. There are three vertical shift conditions that must be differentiated to obtain the correct vertical shift registration for interlaced systems:

1. Vertical field one equals vertical field two;
2. Vertical field one is one less than vertical field two; and
3. Everything else.

Some HRCs shift field one and field two identically, yielding a vertical field one shift that is equal to the vertical field two shift. For HRCs that do not repeat fields or frames (i.e., HRCs that transmit the full frame rate of the video standard), this condition means that what was a field one in the original video stream is also a field one in the processed video stream, and what was a field two in the original is also a field two in the processed.

Other HRCs reframe the video, shifting the sampled frame by an odd number of frame lines. What used to be field one of the original frame becomes field two of the processed frame and what used to be field two of the original becomes the next frame's field one. Visually, the displayed video appears correct since the human cannot perceive a one-line frame shift of the video.

As shown in Figure 6, field one starts with frame line one, and contains all odd-numbered frame lines. Field two starts with frame line zero (topmost frame line), and contains all even-numbered frame lines. For NTSC, field one occurs earlier in time and field two occurs later in time. For PAL, field two occurs earlier in time and field one occurs later in time.

Reframing occurs when either the earlier field moves into the later field and the later field moves into the earlier field of the next frame (one-field delay), or when the later field moves into the earlier field and the earlier field of the next frame moves into the later field of the current frame (one-field advance). For example, when NTSC original field two is moved into the next NTSC frame's field one, the top line of the field moves from original field-two frame line 0 to processed field-one frame line 1. In field line numbering, the top line stays in field line 0, so processed field one has a zero vertical shift (since vertical shifts are measured for each field using field lines). When original NTSC field one is moved to that frame's field two, the top line of the field moves from original field one, frame line 1 to processed field two, frame line 2. In field line numbering, the top line moves from field line 0 to field line 1, so processed field two has a one field line vertical shift. The general rule for both NTSC and PAL is that when the field-two vertical shift (in field lines) is one greater than the field-one vertical shift (in field lines), reframing has occurred.

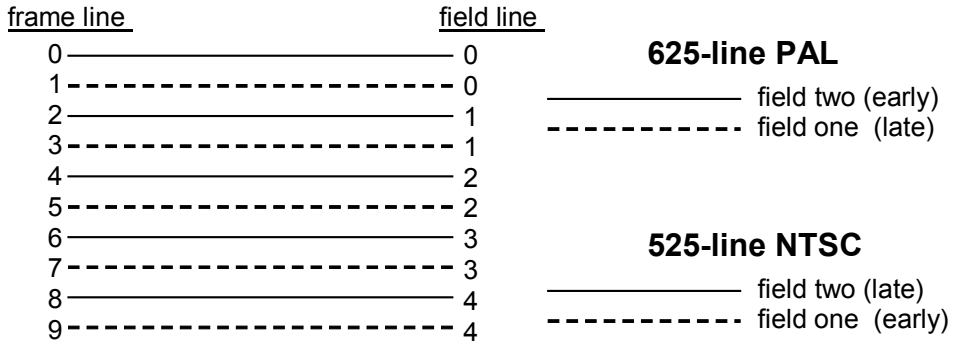

Figure 6 - Diagram depicting interlaced fields and frame/field line numbering scheme

If the field-two vertical shift is not equal to or one more than the field-one vertical shift, the HRC has corrupted the proper spatial sampling of the two interlaced fields of video, and the resulting video will appear to "bob" up and down. Such an impairment is both obvious and annoying to the viewer, and hence seldom occurs in practice since the HRC designer discovers and corrects the error. Therefore, most of the time, spatial registration simplifies into two common patterns:

1. In systems that do not reframe, field-one vertical shift equals field-two vertical shift; and
2. In systems that reframe, field-one vertical shift plus one equals field-two vertical shift.

Additionally, notice that spatial registration includes some temporal registration information; specifically, whether the video has been reframed or not. The temporal registration process may or may not be able to detect reframing, but even if it can, reframing is inherent to the spatial registration process. Therefore, spatial registration must be able to determine whether the processed field being examined best aligns with an original field one or field two. The spatial registration for each field can only be correctly computed when the processed field is compared to the original field that created it. Aside from the

reframing issue, use of the wrong original field (field one versus field two) can produce spatial registration inaccuracies due to the inherent differences in the spatial content of the two interlaced fields.

6.1.3 Required Inputs to the Spatial Registration Algorithm

This section gives a list of the input variables that are required by the spatial registration algorithm. These inputs specify items such as the range of spatial shifts and temporal fields over which to search. If these ranges are overly generous, the speed of convergence of the iterative search algorithm used to find the spatial shift may be slow, and the probability of false spatial registration for scenes with repetitive content is increased (e.g., someone waving their hand). Conversely, if these ranges are too restrictive, the search algorithm will encounter, and *slowly* extend, the search range boundaries with successive iterations. While this built-in search intelligence is useful if the user guesses incorrectly the search uncertainties by a small amount, the undesirable side effect is to dramatically increase run time when the user guesses incorrectly by a large amount. Alternatively, in this case, the search algorithm may fail to find the correct spatial shift.

6.1.3.1 Expected Range of Spatial Shifts

The expected range of spatial shifts for 525-line and 625-line video sampled according to Rec. 601 lies between ±20 *pixels* horizontally and ±12 *field* lines vertically. This range of expected shifts has been determined empirically by processing video data from hundreds of HRCs. The expected range of spatial shifts for video sampled according to other formats smaller than Rec. 601 (e.g., CIF), is presumed to be half of that observed for 525-line and 625-line systems. This search algorithm should operate correctly, albeit a bit slower, when the processed field has spatial shifts that lie outside of the expected range of spatial shifts. This is because the search algorithm will expand the search beyond the expected range of spatial shifts when warranted. Excursions exceeding 50% of the expected range, however, may report a failure to find the correct spatial registration.

6.1.3.2 Temporal Uncertainty

The user must also specify the temporal registration uncertainty, (i.e., the range of original fields to examine for each processed field). This temporal uncertainty is expressed as a number of frames before and after the default temporal registration. If the original and processed video sequences are stored as files, then a reasonable default temporal registration is to assume that the first frames in each file align. The temporal uncertainty that is specified should be large enough to include the actual temporal registration. An uncertainty of plus or minus one second (30 frames for 525-line NTSC video; 25 frames for 625-line PAL video) should be sufficient for most video systems. HRCs with long video delays may require a larger temporal uncertainty. The search algorithm may examine temporal registrations outside of the specified uncertainty range when warranted (e.g., when the farthest original field is chosen as the best temporal registration).

6.1.3.3 Processed Valid Region (PVR) Guess

The PVR guess specifies the portion of the processed image that has not been blanked or corrupted due to processing, presuming no spatial shift has occurred (since the spatial shift has not yet been measured). Although the PVR guess could be determined empirically, a user-specified PVR guess that excludes the over-scan is a good choice. In most cases this will eliminate invalid video from being used in the spatial registration algorithm. For 525-line/NTSC video sampled according to Rec. 601, the over-scan covers approximately 18 frame lines at the top and bottom of the frame, and 22 pixels at the left and right sides of the frame. For 625-line/PAL video sampled according to Rec. 601, the over-scan covers approximately 14 frame lines at the top and bottom of the frame, and 22 pixels at the left and right sides

of the frame. When using other image sizes (e.g., CIF), a reasonable default PVR for these image sizes should be selected.

6.1.4 Sub-Algorithms Used by the Spatial Registration Algorithm

The spatial registration algorithm makes use of a number of sub-algorithms, including estimation of gain and level offset, and the formula used to determine the closest matching original field for a given processed field. These sub-algorithms have been designed to be computationally efficient, since they must be performed many times by the iterative search algorithm.

6.1.4.1 Region of Interest (ROI) Used by All Calculations

All field comparisons made by the algorithm will be between spatially shifted versions of a ROI extracted from the processed video (to compensate for the spatial shifts introduced by the HRC) and the corresponding ROI extracted from the original video. The spatially shifted ROI from the processed video will be denoted as PROI (i.e., processed ROI) and the corresponding ROI from the original video will be denoted as OROI (original ROI). The rectangle coordinates that specify OROI are fixed throughout the algorithm and are chosen to give the largest possible OROI that meets both of the following requirements:

- The OROI must correspond to a PROI that lies within the PVR for all possible spatial shifts that will be examined.
- The OROI is centered within the original image.

6.1.4.2 Gain and Level Offset

The following algorithm is used to estimate the gain of the processed video. The processed field being examined is shift-corrected using the current estimate for spatial shift. After this shift-correction, a PROI is selected that corresponds to the fixed OROI determined in clause 6.1.4.1. Next, the standard deviation of the luminance (Y) pixels from this PROI and the standard deviation of the luminance pixels (Y) from the OROI are calculated. Gain is then estimated as the standard deviation of PROI pixels divided by the standard deviation of OROI pixels.

The reliability of this gain estimate improves as the algorithm iterates toward the correct spatial and temporal shift. A gain of 1.0 (i.e., no gain correction) may be used during the first several iteration cycles. The above gain calculation is sensitive to impairments in the processed video such as blurring. However, for the purposes of spatial registration, this gain estimate is appropriate because it makes the processed video look as much like the original video as possible. To remove gain from the processed field, each luminance pixel in the processed field is divided by the gain.

There is no need to determine or correct for level offset, since the spatial registration algorithm's search criteria is unaffected by level offsets (see clause 6.1.4.3).

6.1.4.3 Formulae Used to Compare PROI with OROI

After correcting the PROI for gain[4] (see 6.1.4.2), the standard deviation of the (OROI-PROI) difference image is used to choose between two or more spatial shifts or temporal shifts. The gain estimate from the previous best match is used to correct the PROI gain. To search among several spatial shifts (with temporal shift held constant), compute the standard deviation of the (OROI-PROI) difference image for

[4] Gain compensation can sometimes be omitted to decrease the computational complexity. However, omission of gain correction is only recommended during early stages of the iterative search algorithm, where the goal is to find the approximate spatial registration (e.g., see clauses 6.1.5.2 and 6.1.5.3).

several PROI generated using different spatial shifts. For a given processed field, the combination of spatial and temporal shifts that produce the smallest standard deviation (i.e., most cancellation with the original) is chosen as the best match.

6.1.5 Spatial Registration Using Arbitrary Scenes

Spatial registration of a processed field from a scene must examine a plurality of original fields and spatial shifts since both the temporal shift (i.e., video delay) and the spatial shift are unknown. As a result, the search algorithm is complex and computationally intense. Furthermore, the scene content is arbitrary, and so the algorithm may find an incorrect spatial registration (clause 6.1.1). Therefore, the prudent course is to compute the spatial registration of several processed fields from several different scenes that have all been passed through the same HRC, and combine the results into one robust estimate of spatial shift. A single HRC should have one constant spatial registration. If not, these time varying spatial shifts would be perceived as an impairment (e.g., the video would bounce up and down or from side to side). This clause describes the spatial registration algorithm from the bottom up: in that the core components of the algorithm are described first, and then their application for spatial registering scenes and HRCs is described.

6.1.5.1 Best Original Field Match in Time

When spatially registering using scene content, the algorithm must find the original field that most closely matches the current processed field. Unfortunately, that original field may not actually exist. For example, a processed field may contain part of two different original fields since it may have been interpolated from other processed fields. The current estimate of the best original field match (i.e., that original field that most closely matches the current processed field) is kept at all stages of the search algorithm.

An initial assumption is made that the first field of the processed Big YUV file aligns with the first field of the original Big YUV file, within plus or minus some temporal uncertainty in frames (denoted here as **U**). For each processed field that is examined by the algorithm, there must be a buffer of **U** original frames before and after this field. Thus, the algorithm starts examining processed fields that are **U** frames into the file, and examines every frequencyth frame thereafter (denoted here as **F**), stopping **U** frames before the end of the file.

The final search results from the previous processed field (gain, vertical and horizontal shift, temporal shift) are used to initialize the search for the current processed field. The best original field match to the current processed field is computed assuming a constant video delay. For example, if processed field **N** was found to best align with original field **M** in the Big YUV files, then processed field **N+F** would be assumed to be best aligned to original field **M+F** at the start of the search.

6.1.5.2 Broad Search for the Temporal Shift

A full search of all possible spatial shifts across the entire temporal uncertainty for each processed field would require a large number of computations. Instead, a multi-step search is used, where the first step is a broad search over a very limited set of spatial shifts, whose purpose is to get close to the correct matching original field.

For the selected processed frame, this broad search examines field one of this frame (see Figure 6) and considers only those original fields of field type one that are spaced two frames apart (i.e., four fields apart) across the entire range of plus and minus the temporal registration uncertainty. The broad search considers the following four spatial shifts of the processed video:

1. No shift;
2. Eight pixels to the left;

3. Eight pixels to the right; and
4. Eight field lines up (see Figure 7).

In Figure 7, positive shifts mean the processed video is shifted down and to the right with respect to the original video. The “eight field lines down” shift is not considered because empirical observations have revealed that very few video systems move the video picture down. The previous best estimate for spatial shift (i.e., from a previously processed field) is also included as a fifth possible shift when it is available. The closest matching original field to the selected processed field is found using the comparison technique described in clause 6.1.4.3. The temporal shift implied by the closest matching original field becomes the starting point for the next step of the algorithm, a broad search for the spatial shift (clause 6.1.5.3). According to the coordinate system in Figure 4, a positive temporal shift means that the processed video has been shifted in the positive time direction (i.e., the processed video is delayed with respect to the original video). With respect to the original and processed Big YUV files, a positive field shift thus means that fields must be discarded from the beginning of the processed Big YUV file while a negative field shift means that fields must be discarded from the beginning of the original Big YUV file.

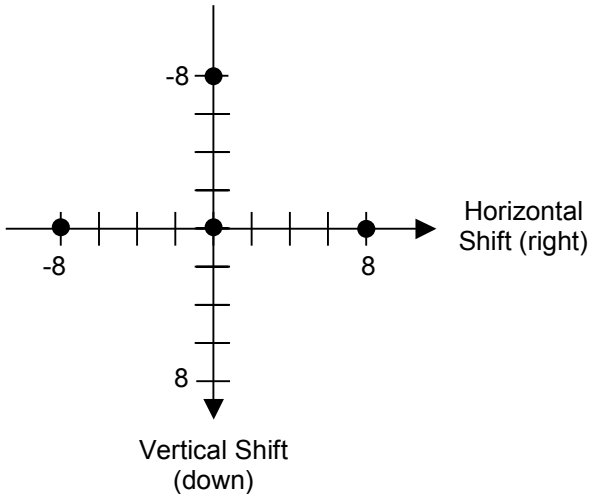

Figure 7 - Spatial shifts considered by the broad search for the temporal shift

6.1.5.3 Broad Search for the Spatial Shift

Using the temporal registration found by the broad search for temporal shift (clause 6.1.5.2), a broad search for the correct spatial shift is now performed using a more limited range of original fields. The range of original fields that are considered for this search include the best matching original field of field type one (from clause 6.1.5.2) and the four next closest original fields that are also of field type one (field type ones from the 2 frames before and after the best matching original field). The broad search for spatial shift covers the range of spatial shifts given in Figure 8. Notice that fewer downward shifts are considered (as in clause 6.1.5.2), since these are less likely to be encountered in practice. The set of spatial shifts and original fields is searched using the comparison technique described in clause 6.1.4.3. The resulting best temporal and spatial shifts now become the improved estimates for the next step of the algorithm given in clause 6.1.5.4.

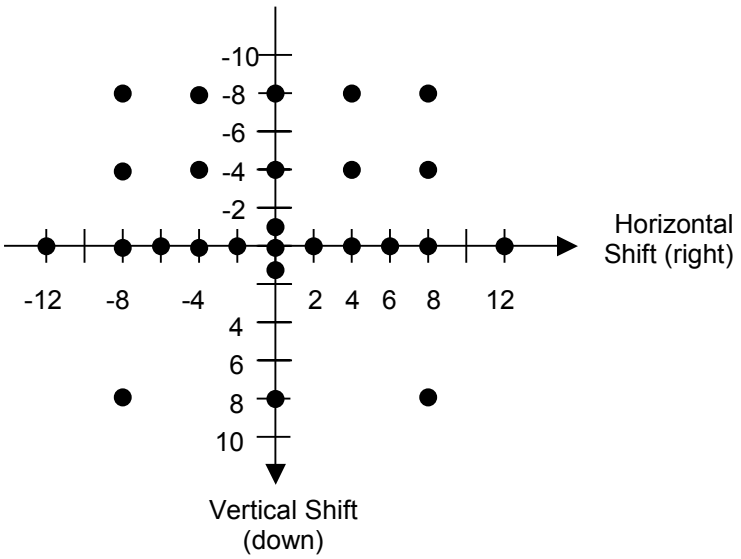

Figure 8 - Spatial shifts considered by the broad search for the spatial shift

6.1.5.4 Fine Search for the Spatial-Temporal Shift

The fine search includes a much smaller set of shifts centered around the current spatial registration estimate and just five fields centered around the current best matching original field. Thus, if the best matching original field were a field type one, the search would include three field type ones and the two field type twos. The spatial shifts that are considered include:

- The current shift estimate;
- All eight shifts that are within one pixel or one line of the current estimate;
- Eight shifts that are two pixels or two lines from the current shift estimate; and
- The zero shift condition (see Figure 9).

In the example shown in Figure 9, the current spatial shift estimate for the processed video is a shift of 7 field lines up and 12 pixels to the right of the original video. The set of spatial shifts shown in Figure 9 form a near-complete local search of the spatial registrations near the current spatial registration estimate. The zero shift condition is included as a safety check that helps prevent the algorithm from wandering and converging to a local minimum. The set of spatial shifts and original fields is thoroughly searched using the comparison technique described in clause 6.1.4.3. The resulting best temporal and spatial shifts now become the improved estimates for the next step of the algorithm given in clause 6.1.5.5.

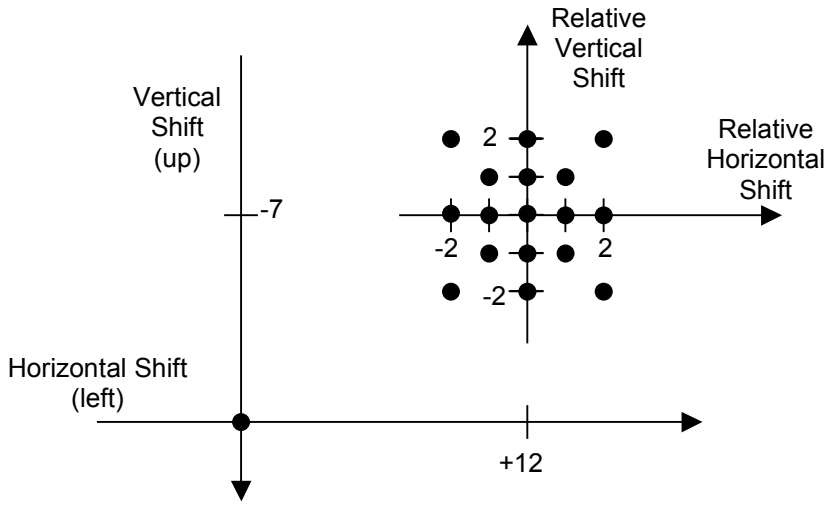

Figure 9 - Spatial shifts considered by the fine search for the spatial shift

6.1.5.5 Repeated Fine Searches

Iteration through the fine search of clause 6.1.5.4 will move the current estimate for spatial shift a little closer to either the actual spatial shift or (more rarely) a false minimum. Likewise, one iteration through the fine search will move the current estimate for the best-aligned original field either a little closer to the actual best-aligned original field or (more rarely) a little closer to a false minimum. Thus, each fine search will move these estimates closer to a stable value. Because fine searches examine a very limited area spatially and temporally, they must be performed repetitively to assure that convergence has been reached. When gain compensation is being used, the processed field's gain is estimated anew between each fine search (see clause 6.1.4.2).

Repeated fine searches are performed on the processed field (see 6.1.5.4) until the best spatial shift *and* the original field associated with that spatial shift remain unchanged from one search to the next. Repeated fine searches are stopped if the algorithm is alternating between two spatial shifts (e.g., a horizontal shift 3 and then a horizontal shift 4, with everything else remaining the same). This alternation is indicated when the current best estimate for spatial shift *and* the original field associated with that spatial shift are identical to those found two iterations ago.

Sometimes the repeated search algorithm fails to converge. If the algorithm fails to converge within some requested maximum number of iterations, the iterative search algorithm is terminated and a "failure to find shift" condition is reported for that processed field. This special case does not normally pose a problem because multiple processed fields are examined for each scene (see 6.1.5.6) and multiple scenes are examined for each HRC (see 6.1.5.7).

6.1.5.6 Algorithm for One Scene

An initial baseline (i.e., starting) estimate for vertical shift, horizontal shift, and temporal registration is computed without any gain compensation as follows. The first temporal uncertainty (**U**) processed frames in the Big YUV file are skipped. A broad search for the temporal shift is performed on the next processed field of field type one (see 6.1.5.2). Notice that this broad search will search the first **U***2 + 1 frames of the original video sequence for a field type one that best aligns. Then, a broad search for the spatial shift is performed centered on this best-aligned original field (see 6.1.5.3). Next, perform up to five fine spatial-temporal searches to fine-tune the spatial and temporal estimates (see 6.1.5.4 and 6.1.5.5). If these repeated fine searches fail to find a stable result, discard this processed field from consideration.

Repeat the above procedure every frequencyth (F) frame until an original field of field type one is found that produces stable results. The baseline estimate will be updated periodically, as described below.

The spatial shift estimates are calculated for both field types of a frame in the processed Big YUV file as follows. Using the baseline estimate as a starting point, perform up to three repeated fine searches on the first processed field of field type one. If the baseline estimate is correct or very nearly correct, the repeated fine searches will yield a stable result. If so, the spatial shift and temporal delay for that processed field are stored in an array that is dedicated to storing the field one results. If a stable result is not found, most likely the spatial shift is correct but the temporal shift estimate is off (i.e., the current estimate of temporal shift is more than two frames away from the true temporal shift). So a broad search for the temporal shift is conducted that includes the current best estimate of spatial shift. This broad search will normally correct the temporal delay estimate. When the broad search for the temporal shift completes, its output is used as the starting point, and up to five repeated fine searches are performed. If this second repeated fine search fails to find a stable result, then report a failed spatial registration for this frame (i.e., both field type one and field type two). If a stable result is found from this second search, then the spatial shift and temporal delay for that field are stored in the field one array. Also, the spatial shift and temporal delay used as the starting point for the next processed field of field type one are updated (i.e., for the first processed field, the baseline results are used and after that, the last stable result is used). After the spatial shift has been estimated for the first processed field of field type one, the spatial shift for the first processed field of field type two is estimated. Using the field one spatial results as the starting point, the same steps are used to find the field-two spatial shift (i.e., the three fine searches, and if needed a broad search for the temporal shift followed by five repeated fine searches). If a stable result is found for field two, store the vertical and horizontal shift for field two in a different array that is dedicated to storing field-two results.

The procedure described in the above paragraph is applied to estimate the spatial shift of both field types of each frequencyth (**F**) frame in the Big YUV file that contains the processed video. The first temporal uncertainty (**U**) processed frames in the Big YUV file are skipped. This sequence of estimates is then used to compute robust estimates of the spatial shift for each field type for the scene being examined. The vertical field-one shift results from each frame are sorted, and the 50th percentile retained as the overall vertical field-one shift. Likewise, the vertical field-two shift results from each frame are sorted, and the 50th percentile retained as the overall vertical field-two shift. The horizontal field-one shift results from each frame are sorted, and the 50th percentile retained as the overall horizontal shift. Any difference between field-one and field-two horizontal shift is most likely due to a sub-pixel horizontal shift (e.g., a horizontal shift of 0.5 pixels). Sub-pixel horizontal shifts will produce estimates that include both of the two closest shifts. Using the 50th percentile point allows the most likely horizontal shift to be chosen, which produces a spatial registration accuracy that is good to the nearest 0.5 pixels.[5]

6.1.5.7 Algorithm for One HRC

If several scenes have been passed through the same HRC, the spatial registration results for each scene should be identical. Thus, filtering results obtained from multiple scenes can increase the robustness and accuracy of the spatial shift measurements. The overall HRC spatial registration results can then be used to compensate all of the processed video for that HRC.

6.1.5.8 Comments on Algorithm

Some video scenes are simply not well suited for estimating spatial registration. The described algorithm will sometimes locate a false minimum. Other times, the algorithm will wander between multiple solutions and never reach a stable result. For these reasons, it is advisable to examine multiple images within the same scene and to median filter (i.e., sort results from low to high and select the 50th percentile point)

[5] Spatial registration to the nearest 0.5 pixels is sufficient for the video quality measurements described in this standard. Sub-pixel spatial registration techniques are beyond the scope of this standard.

these results across different scenes. The spatial registration by scenes algorithm is a heuristic algorithm that utilizes patterns of spatial shifts that have been observed from a sampling of video systems. These assumptions may be incorrect for some systems, causing the algorithm to find an incorrect spatial shift. However, failure of the algorithm tends to produce spatial shifts that are inconsistent from frame to frame and from scene to scene (i.e., when the algorithm fails, it normally produces a scattering of results). When the algorithm outputs the same or very similar spatial shifts for each scene, a high degree of confidence is indicated. When the individual field results for a scene wander, a low degree of confidence is indicated.

6.1.6 Spatial Registration of Progressive Video

Spatial registration of progressive video follows the same steps as the interlace algorithms, with minor modifications. Where the interlace algorithms operate on field one and field two separately, the progressive algorithm operates on frames. Thus, all mentions of field two are ignored and -- with the exception of the fine searches -- the range of vertical shifts is doubled.

The modification of the vertical shift range is most important for the broad spatial shift. When doing a broad search for spatial shift (see 6.1.5.3) the numbers on the vertical axis in Figure 8 must be doubled (e.g., +8 becoming +16 and –4 becoming –8).[6] In addition, for progressive CIF and QCIF images, the horizontal and vertical broad spatial search ranges are halved due to the smaller shifts that are typically encountered with these image sizes. For example, using CIF images in Figure 8, the horizontal axis would stretch from –6 to +6 pixels and the vertical axis would stretch from –8 to +8 frame lines.

The temporal search range, being stated in frames, is largely unchanged. For the broad temporal search in clause 6.1.5.2, instead of matching one processed field one to every second original field one, the progressive algorithm compares one processed frame to every second original frame. For the colorbar algorithm, the search examines spatial shifts for one processed frame and one original frame (i.e., no temporal searching).

The only step requiring more complicated changes is the fine search from clause 6.1.5.4. Here, the vertical shifts remain unchanged, lying between -2 frame lines and +2 frame lines. Thus, the vertical axis of Figure 9 is interpreted as referring to frame lines. The temporal extent of this fine search may be set to five original frames centered on the current aligned original frame, instead of the three original frames otherwise implied. A five-frame search extent may improve the speed and efficiency of the fine search when compared to the interlace version of the algorithm, since progressive HRCs are more likely to contain varying video delay than non-zero spatial shifts.

When considering the algorithmic changes for progressive video systems, many of the spatial shift search parameters can be modified without harming the integrity of the algorithm. As an example, consider spatial shifts other than zero pixels and zero lines for the broad temporal search. The spatial shift at zero pixels horizontally and 8 field lines vertically for interlace video could be moved to 16 frame lines for progressive video (as recommended above) or placed at 8 frame lines, under the assumption that progressive video sequences are unlikely to contain 16 frame lines of vertical shift. Likewise, spatial shift at zero lines vertically and 8 pixels horizontally could be moved to 9 or 10 pixels horizontally without any detrimental effects. As another example, the exact number of repeated fine searches performed could be increased or decreased for specific applications. The exact values recommended here are significantly less important than the actual structure of the search algorithm.

[6] In one possible exception to this doubling, the spatial shift associated with zero pixels horizontally and plus or minus one field line vertically could be left at plus or minus one frame line vertically. Spatial shifts very close to (zero, zero) are commonly encountered.

6.2 Valid Region

NTSC (525-line) and PAL (625-line) video sampled according to Rec. 601 may have a border of pixels and lines that do not contain a valid picture. The original video from the camera may only fill a portion of the Rec. 601 frame. A digital video system that utilizes compression may further reduce the area of the picture in order to save transmission bits. If the non-transmitted pixels and lines occur in the over-scan area of the television picture, the typical end-user should not notice the missing lines and pixels. If these non-transmitted pixels and lines exceed the over-scan area, the viewer may notice a black border around the picture, since the system will normally insert black into this non-transmitted picture area. Video systems (particularly those that perform low-pass filtering) may exhibit a ramping up from the black border to the picture area. These transitional effects most often occur at the left and right sides of the image but can also occur at the top or bottom. Occasionally, the processed video may also contain several lines of corrupted video at the top or bottom of the picture that may not be visible to the viewer (e.g., VHS tape recorders corrupt several lines at the bottom of the picture in the over-scan area). To prevent non-picture areas from influencing the VQM measurements, these areas should be excluded from the VQM measurement. The automated valid region algorithm presented here estimates the valid region of the original and processed video streams so that subsequent computations do not consider corrupted lines at the top and bottom of the Rec. 601 frame, black border pixels, or transitional effects where the black border meets the picture area.

6.2.1 Core Valid Region Algorithm

This section describes the core valid region algorithm that is applied to a single original or processed image. This algorithm requires three input arguments:

1. **Image**. The core algorithm uses the Rec. 601 luminance image of a single video frame. When measuring the valid region of a *processed* video sequence, any spatial shift imposed by the video system must have been removed from the luminance image before applying the core algorithm (see spatial registration in 6.1).
2. **Maximum Valid Region**. The core algorithm will not consider pixels and lines outside of a maximum valid video region. This provides a mechanism for the user to specify a maximum valid region that is smaller than the entire image area if *a priori* knowledge indicates that the sampled image has corrupted pixels or lines as discussed in 6.2.
3. **Current Valid Region**. The current valid region is an estimate of the valid region and lies entirely within the maximum valid region. All pixels inside the current valid region are known to contain valid video; pixels outside the current valid region may or may not contain valid video content. Initially, the current valid region is set to the smallest possible area located at the exact center of the image.

The core algorithm examines the area of video between the maximum valid region and the current valid region. If some of those pixels appear to contain valid video, the current valid region estimate is enlarged. The algorithm will now be described in detail for the left edge of the image.

1. Compute the mean of the left-most column of pixels in the maximum valid region. The left-most column of pixels will be denoted as column “J-1” and the mean will be represented by “M_{J-1}”.
2. Take the mean of the next column of pixels, “M_J”.
3. Column J is declared invalid video if it is black, ($M_J < 20$) or if the average pixel level of the mean value for successive columns indicates a ramp up from black border to valid picture ($M_J - 2 > M_{J-1}$). If either of these conditions are true, increment J and repeat steps (2) and (3). Otherwise, go to step (4).
4. If final column J is within the current valid region, then no new information has been obtained. Otherwise, update the current valid region with J as the left coordinate.

The algorithm for finding the top edge of the image is similar to that given above for the left edge. For the bottom and right edges, J is decremented instead of incremented; otherwise the algorithm is the same. The values produced for top, left, bottom, and right indicate the last valid pixel or line.

The stopping conditions identified in step (3) can be fooled by scene content. For example, an image that contains genuine black at the left side (i.e., black that is part of the scene) will cause the core algorithm to conclude that the left-most valid column of video is farther toward the middle of the image than it ought to be. For that reason, the core algorithm is applied to multiple images from a video sequence, thereby increasing the accuracy of the valid region estimate.

6.2.2 Applying the Core Valid Region Algorithm to a Video Sequence

6.2.2.1 Original Video

The core algorithm is first applied to the original sequence of images. For NTSC video sampled according to Rec. 601 (see clause 5), the recommended setting for the maximum valid region is top = 6, left = 6, bottom = 482, right = 714. For PAL video sampled according to Rec. 601, the recommended setting for the maximum valid region is top = 6, left = 16, bottom = 570, right = 704. The core algorithm is run on the first image in the video sequence, and every frequencyth image thereafter. For example, if the specified frequency were 15, the core algorithm would examine sequence image numbers 0, 15, 30, 45, and so forth. When all images in the sequence have been examined, the current valid region will contain the largest valid area implied by any examined image in the video sequence. Pixels and lines between this final current valid region and the maximum valid region are considered to contain either black or a transitional ramp up from black.

The final valid region must contain an even number of lines and an even number of pixels. Any odd top or left coordinates are incremented by one. Then, if the region contains an odd number of lines, bottom is decremented; likewise, if the region contains an odd number of pixels (e.g., horizontally), right is decremented. This simplifies color processing for video sampled in accordance with Rec. 601, since the color channels are sub-sampled by 2 when compared to the luminance channel. Also, each interlaced field of video will contain the same number of video lines. This will assure that spatial-temporal sub-regions (from which features will be extracted) always contain valid video with equal contributions from both interlaced fields. The resulting valid region is returned as the original valid region.

6.2.2.2 Processed Video

When computing the valid region of the processed video sequence, the maximum valid region setting for the core algorithm is first set equal to the corresponding original valid region found for that scene. This maximum valid region is then reduced in size by any pixels and lines that are considered invalid due to spatially shift correcting the processed video frames. The core algorithm is then run on the first image in the processed video sequence, and every frequencyth image thereafter (i.e., if frequency = F, use images **Y**(0), **Y**(F), **Y**(2F), **Y**(3F), and so forth).

After the core algorithm has been applied to the processed video sequence, the valid region found by the core algorithm is reduced inward by a safety margin. The recommended safety margin discards one line off the top and bottom, and five pixels off the left and right. The large left and right safety margins ensure that any ramp up or down from black is excluded from the processed valid region.

The final processed valid region must contain an even number of lines and an even number of pixels. Any odd top or left coordinates are incremented by one. Then, if the region contains an odd number of lines, bottom is decremented; likewise, if the region contains an odd number of pixels (e.g., horizontally), right is decremented. The resulting valid region is returned as the processed valid region.

6.2.3 Comments on Valid Region Algorithm

This automated valid region algorithm will work well to estimate the valid region of most scenes. Due to the nearly infinite possibilities for scene content, the algorithm described herein takes a conservative approach to estimation of the valid region. A manual examination of valid region would quite likely choose a larger region. Conservative valid region estimates are more suitable for an automated video quality measurement system, because discarding a small amount of video will have little impact on the quality estimate and in any case this video usually occurs in the over-scan part of the video. On the other hand, including corrupted video in the video quality calculations may have a large impact on the quality estimate.

This algorithm does not contain sufficient artificial intelligence to distinguish between corrupted pixels and lines at the edge of an image and true scene content. A rule of thumb is used instead, stating that such invalid video generally occurs at the extreme edges of the image. Specification of a conservative user-definable maximum valid video region (i.e., the starting point for the automated algorithm) provides a mechanism to exclude these possibly corrupt image edges from consideration.

When the valid region algorithm is applied to video that is not sampled according to Rec. 601 (e.g., the CIF, used by ITU-T Recommendation H.261), the recommended setting for maximum valid region when examining the original video is the entire image. In these cases, the sampled video does not normally include any corrupted over-scan, so a maximum valid region smaller than the entire image is unnecessary.

6.3 Gain and Offset

6.3.1 Core Gain and Level Offset Algorithm

This section explains the method for performing gain and level offset calibration. A prerequisite before applying this algorithm is that the original and processed images be spatially registered (see 6.1). The original and processed images must also be temporally registered, which will be addressed later in 6.4. Gain and level offset calibration can be performed on either fields or frames as appropriate.

The method presented here makes the assumption that the Rec. 601 Y, C_B, and C_R signals each have an independent gain and level offset. This assumption will, in general, be sufficient for calibrating component video systems (e.g., Y, R-Y, B-Y). However, in composite or S-video systems, it is possible to have a phase rotation of the chrominance information since the two chrominance components are multiplexed into a complex signal vector with amplitude and phase. The algorithm presented here will not properly calibrate video systems that introduce a phase rotation of the chrominance information (e.g., the hue adjustment on a television set).

As previously noted, this calibration model assumes that there is no cross coupling between any of the three video components. With this assumption, the core calibration algorithm is applied independently to each of the three channels: Y, C_B, and C_R.

The valid region of the original and processed image plane is first divided into N sub-regions. For each of the sub-regions, the mean *original* and *processed* values are computed (i.e., mean over space). Next, these *original* and *processed* values are represented as N-element column vectors $\underline{O}$ and $\underline{P}$, respectively:

$$\underline{O}_{N\text{x}1} = \begin{bmatrix} original_1 \\ . \\ . \\ . \\ original_N \end{bmatrix}, \quad \underline{P}_{N\text{x}1} = \begin{bmatrix} processed_1 \\ . \\ . \\ . \\ processed_N \end{bmatrix}.$$

Calibration involves computing the gain (*g*) and level offset (*l*) according to the following model:

$\underline{P} = g\underline{O} + l$.

Since there are only two unknowns (i.e., *g* and *l*) but N equations (i.e., N sub-regions), we must solve the over-determined system of linear equations given by:

$$\hat{\underline{P}} = A \begin{bmatrix} l \\ g \end{bmatrix},$$

where *A* is an N x 2 matrix given by $A_{N\text{x}2} = [\underline{1} \quad \underline{O}]$, and $\underline{1}$ is an N-element column vector of '1s' given by

$$\underline{1}_{N\text{x}1} = \begin{bmatrix} 1_1 \\ . \\ . \\ . \\ 1_N \end{bmatrix}.$$

$\hat{P}$ is the estimate of the processed samples if the gain and level offset correction were applied to the original samples. The least squares solution to this over-determined problem (provided N > 2) is given by

$$\begin{bmatrix} l \\ g \end{bmatrix} = \left(A^{\mathrm{T}} A\right)^{-1} A^{\mathrm{T}} P,$$

where the superscript "T" denotes matrix transpose and the superscript "-1" denotes matrix inverse.

When the core gain and offset algorithm is independently applied to each of the three channels, six estimates result: Y gain, Y offset, C_B gain, C_B offset, C_R gain, and C_R offset.

6.3.2 Using Scenes

The basic algorithm given in 6.3.1 can be applied to original and processed video streams provided they have been spatially and temporally registered. This scene-based technique divides the image into abutting blocks with unknown intensity levels. A sub-region size of 16 lines x 16 pixels is recommended for frames (i.e., 8 lines x 16 pixels for one Y NTSC or PAL field; 8 lines x 8 pixels for C_B and C_R due to sub-sampling of the color image planes). The mean over space of the [Y, C_B, C_R] samples is computed for each corresponding original and processed sub-region, or block, to form a spatially sub-sampled image. All the selected blocks must lie within the PVR.

6.3.2.1 Registering the Processed Images

For simplicity, we will assume that the best spatial registration has already been found using one of the techniques presented in 6.1. Before gain and level offset are estimated, each processed image must also be temporally registered. The original image that best aligns with the processed image must be used for the gain and level offset calculation. If the video delay is variable, this temporal registration must be performed for each processed image. If the video delay is constant for the scene, the temporal registration only needs to be performed once.

To temporally register a processed image, first create the spatially sub-sampled original and processed fields (or frames for progressive video) as specified in 6.3.2, after correcting for the spatial shift of the processed video. Using the sub-sampled Y images, apply the search function given in 6.1.4.3, except perform this search using all the original images that are within the temporal registration uncertainty (**U**). Use the best resulting temporal registration for all three image planes, Y, C_B, and C_R.

6.3.2.2 Gain & Level Offset of Registered Images

An iterative least squares solution with a cost function is used to help minimize the weight of outliers in the fit. This is because outliers are normally due to distortions rather than pure level offset and gain changes, and assigning equal weight to these outliers would distort the fit.

The following algorithm is applied separately to the N matching original and processed pixels from each of the three spatially sub-sampled images [Y, C_B, C_R].

1. Use the normal least squares solution from 6.3.1 to generate the initial estimate of the level offset and gain: $\begin{bmatrix} l \\ g \end{bmatrix} = \left(A^{\mathrm{T}} A\right)^{-1} A^{\mathrm{T}} \underline{P}$.
2. Generate an error vector (*E*) that is equal to the absolute value of the difference between the true processed samples and the fitted processed samples: $\underline{E} = \left|\underline{P} - \underline{\hat{P}}\right|$.
3. Generate a cost vector (*C*) that is the element-by-element reciprocal of the error vector (*E*) plus a small epsilon (ε): $\underline{C} = \dfrac{1}{\underline{E} + \varepsilon}$. The ε prevents division by zero and sets the relative weight of a point that is on the fitted line versus the weight of a point that is off the fitted line. An ε of 0.1 is recommended.
4. Normalize the cost vector *C* for unity norm (i.e., each element of *C* is divided by the square root of the sum of the squares of all the elements of *C*).
5. Generate the cost vector $\underline{C}^2$ that is the element-by-element square of the cost vector *C* from step 4.
6. Generate an N x N diagonal cost matrix (C^2) that contains the cost vector's elements ($\underline{C}^2$) arranged on the diagonal, with zeros everywhere else.
7. Using the diagonal cost matrix (C^2) from step 6, perform cost-weighted least squares fitting to determine the next estimate of the level offset and gain: $\begin{bmatrix} l \\ g \end{bmatrix} = \left(A^{\mathrm{T}} C^2 A\right)^{-1} A^{\mathrm{T}} C^2 \underline{P}$.
8. Repeat steps 2 through 7 until the level offset and gain estimates converge to four decimal places.

These steps are applied separately to processed field one and processed field two, to obtain two estimates for *g* and *l*. Field one and two must be examined separately, because the temporally registered original fields need not correspond to one frame within the original video sequence. For progressive video, the above steps are applied to the entire processed frame at once.

6.3.2.3 Estimating Gain and Level Offset for a Video Sequence and HRC

The algorithm described above is applied to multiple matching original and processed field pairs distributed every frequencyth frame throughout the scene (for progressive video, original and processed frame pairs). A median filter is then applied to the six time histories of the level offsets and gains to produce average estimates for the scene.

If several scenes have been passed through the same HRC, the level offset and gain for each scene will be considered to be identical. Thus, filtering results obtained from multiple scenes can increase the

robustness and accuracy of the level offset and gain measurements. The overall HRC level offset and gain results can then be used to compensate all of the processed video for that HRC.

6.3.3 Applying Gain and Level Offset Corrections

The temporal registration algorithms (see 6.4) and most quality features (see clause 7) will specify that the gain calculated herein should be removed. To remove gain and level offset from the Y plane, apply the following formula to each processed pixel:

New Y(i,j,t) = [Y(i,j,t) – *l*] / *g*

Gain and level offset correction is not performed on the color planes (i.e., C_B and C_R). Perceptual chrominance errors are instead captured by the color metrics. The C_B and C_R image planes may be gain and level offset corrected for display purposes.

6.4 Temporal Registration

Modern digital video communication systems typically require several tenths of a second to process and transmit the video from the sending camera onto the receiving display. Excessive video delays impede effective two-way communication. Therefore, objective methods for measuring end-to-end video communications delay are important to end-users for specification and comparison of services and to equipment / service providers to optimize and maintain their product offerings. Video delay can depend upon dynamic attributes of the original scene (e.g., spatial detail, motion) and video system (e.g., bit-rate). For instance, scenes with large amounts of motion can suffer more video delay than scenes with small amounts of motion. Thus, video delay measurements should be made in-service to be truly representative and accurate. Estimates of video delay are required to temporally align the original and processed video features shown in Figure 2 before making quality measurements.

Some video transmission systems may provide time synchronization information (e.g., original and processed frames may be labeled with some kind of a frame numbering scheme). In general, however, time synchronization between the original and processed video streams must be measured. This clause presents a technique for estimating video delay based upon the original and processed video frames. The technique is “frame-based” in that it works by correlating lower resolution images, sub-sampled in space and extracted from the original and processed video streams. This frame-based technique estimates the delay of each frame or field (for interlaced video systems). These individual estimates are combined to estimate the average delay of the video sequence.

6.4.1 Frame-Based Algorithm for Estimating Variable Temporal Delays between Original and Processed Video Sequences

This section describes a frame-based temporal registration algorithm. To reduce the influence of distortions on temporal registration, images are spatially sub-sampled and normalized to have unit variance. This algorithm temporally registers each processed image separately, locating the most similar original image. Some of these individual temporal registration measurements may be incorrect, but those errors will tend to be randomly distributed. When delay measurements from a series of images are combined by means of a voting scheme, the overall estimate for the average delay of a video sequence becomes quite accurate. This temporal registration algorithm does not use still and nearly motionless portions of the scene, since the original images are nearly identical to each other.

6.4.1.1 Constants Used by the Algorithm

- BELOW_WARN: Threshold used when examining correlations for deciding if a secondary correlation maximum is sufficiently large so as to indicate ambiguous temporal registration. A BELOW_WARN of 0.9 is recommended.

- BLOCK_SIZE: The sub-sampling factor. Specified in frame lines vertically and pixels horizontally. A BLOCK_SIZE of 16 is recommended.
- DELTA: Secondary maximums in the correlation curve that are within DELTA of the maximum (best) correlation are ignored. A DELTA of 4 is recommended.
- HFW: Half of the filter width for the filter used to smooth the histogram of frame-by-frame temporal registration values. A HFW of 3 is recommended.
- STILL_THRESHOLD: A threshold that is used to detect still video scenes (frame-based temporal registration cannot be used on still video scenes). A STILL_THRESHOLD of 0.002 is recommended.

6.4.1.2 Inputs to the Algorithm

A sequence of N original video luminance images: $\mathbf{Y_O}(t)$, $0 \le t < N$. [7]

A sequence of N processed video luminance images: $\mathbf{Y_P}(t)$, $0 \le t < N$.

Gain and offset correction factors for the processed luminance images.

Spatial registration information: horizontal shift and vertical shift. For interlace video, the vertical shift for each field determines whether the processed video requires reframing.

Valid region of the processed video sequence (i.e., PVR).

Uncertainty (U): a number indicating the accuracy of the initial temporal registration. The initial temporal registration assumption is that the true temporal registration for $\mathbf{Y_P}(t)$ is within plus or minus (U – HFW) of $\mathbf{Y_O}(t)$, for all $0 \le t < N$.

6.4.1.3 Frames versus Fields

The frame-based temporal registration algorithm works for both progressive and interlace video. If the video sequence is progressive, the algorithm aligns frames. If the video sequence is interlaced, the algorithm aligns fields. When aligning interlaced video sequences, either frame or reframed alignments are considered but not both. When frame alignments are considered, field one of the processed video is compared to field one of the original video, and field two of the processed video is compared to field two of the original video. When reframed alignments are considered, field one of the processed video is compared to field two of the original video, and field two of the processed video is compared to field one of the original video. The spatial registration values that are input to the algorithm determine whether frame or reframe alignments are considered. The presence of reframing is detected by examining the vertical spatial registration for each field. If the field one vertical shift equals the field two vertical shift, then the processed video is not reframed; only frame alignments are considered. If the field two vertical shift is one greater than the field one vertical shift, only reframe alignments are considered. All other combinations of vertical shifts indicate problems that should be fixed prior to temporal registration.

6.4.1.4 Description of the Algorithm

1. **Calibrate the video sequences:** Correct the processed video sequence, $\mathbf{Y_P}(t)$, using the spatial registration and gain-offset information given as inputs to the algorithm.

[7] When interlace video requires reframing, the lengths of the original and processed video sequences must be reduced by one to accommodate the reframing. This will reduce the length of the file by one video frame from N as specified in Figure 3.

2. **Select the sub-region of video to be used:** The sub-region of interest to be used by the algorithm must be a multiple of the BLOCK_SIZE and must fit within the PVR. The largest sub-region that meets these two requirements and is closest to the center of the image should be selected. All further processing will be limited to video within this selected sub-region of interest.

3. **Spatially sub-sample the original and processed images:** Spatially sub-sample the region of interest of $\mathbf{Y_O}(t)$ and $\mathbf{Y_P}(t)$ by a factor of BLOCK_SIZE by computing the mean of each block. For progressive video frames, the sub-sampling will be BLOCK_SIZE horizontally and vertically, while for interlace video fields, the sub-sampling will be BLOCK_SIZE horizontally and BLOCK_SIZE/2 vertically. For example, sub-sampling a progressive video sequence by a BLOCK_SIZE of 16 will take the mean of each 16 pixel by 16 frame line block, while sub-sampling an interlace video sequence by a BLOCK_SIZE of 16 will take the mean of each 16 pixel by 8 field line block. This sub-sampling reduces the impact of impairments on the temporal registration process.

4. **Normalize the sub-sampled images:** Normalize each sub-sampled image by the standard deviation of that image. Skip this normalization for any image where the standard deviation is less than one (e.g., images containing a flat field of color).[8] This normalization will minimize the influence of fluctuations in individual image contrast and energy from influencing the temporal registration results. After this step, the original video and processed video sequences will be denoted as $\mathbf{S_O}(t)$ and $\mathbf{S_P}(t)$, respectively, to denote that the images have been sub-sampled and normalized.

5. **Compare processed images to original images:** Compare each processed image, $\mathbf{S_P}(t)$, with the original images $\mathbf{S_O}(t+d)$, where the valid values of d are: $(-U \le d \le +U)$ and the valid values of t are: $(U \le t < N - U)$. For processed image t and original image $t+d$, these comparisons will be denoted as $\mathbf{C}_{td}$ and are computed as the standard deviation over space of the image formed by subtracting processed image t from original image $t+d$: $\mathbf{C}_{td}=std_{\text{space}}(\mathbf{S_O}(t+d)-\mathbf{S_P}(t))$. These comparisons, $\mathbf{C}_{td}$, correlate the t^{th} processed image with each original image that is within the registration uncertainty. Lower values of $\mathbf{C}_{td}$ indicate that the processed image looks more like the original image since more of the image variance is cancelled. The range for t, $U \le t < N - U$, covers all processed images for which original images are available for the entire range of temporal registration uncertainty.

6. **Perform an overall check for still video:** To determine if there is sufficient motion in the video sequence, average $\mathbf{C}_{td}$ over time index t for each d:

$$A_d = \frac{1}{N - 2 * U} \sum_{t=U}^{N-U-1} \mathbf{C}_{td} .$$

   This summation includes the range of processed video images t for which the full uncertainty of original images is available. A_d contains one value for each temporal registration delay d being considered. If (maximum(A_d) - minimum(A_d) < STILL_THRESHOLD), then the scene contains insufficient motion for frame-based temporal registration. The entire scene is still or nearly still. The correlation results from the different video delays are then so similar that any differentiation will be due to random chance rather than reliable measurements. If a still video sequence is detected, the user is given a warning to that effect and the algorithm exits at this point.

7. **Temporally register each processed image:** For each processed image t $(U \le t < N - U)$, find the d within the temporal uncertainty $(-U \le d \le +U)$ that minimizes $\mathbf{C}_{td}$. In other words, for each processed image t, find $d_{\min}(t)$ such that $\mathbf{C}_{t\ d\min(t)} \le \mathbf{C}_{td}$, for all d. The best temporal registration of processed image t is given by $d_{\min}(t)$. Most of the time, the temporal registration indicated for individual images will be correct or very close to correct. The temporal registration will be

[8] Normalization is skipped when the standard deviation is less than one to prevent amplification of noise and to prevent the possibility of dividing by zero for images that contain a flat or uniform intensity level.

incorrect for some images due to various reasons (image distortion, errors, noise, insufficient motion, etc.).

8. **Perform a stillness check on each processed image:** If for a given processed image *t* and all values of *d* (-U ≤ *d* ≤ U), maximum($\mathbf{C}_{td}$) - minimum($\mathbf{C}_{td}$) < STILL_THRESHOLD, then $d_{min}(t)$ is undefined for this processed image *t*. Specifically, there is insufficient motion around image *t* for frame-based temporal registration to work properly.

9. **Form a histogram of all defined temporal registrations:** Compute a histogram using all the defined values of $d_{min}(t)$ with 2*U+1 bins where each bin represents a different video delay (i.e., from –U to +U). Values of $d_{min}(t)$ that are undefined (e.g., still images) are left out of the histogram calculation. This histogram, denoted by H_d, is the histogram of temporal delays for all the processed images that had sufficient motion to perform valid temporal registration. Each bin in the histogram contains the number of processed images with that video delay *d*, where *d* can take values from –U to +U.

10. **Form a smoothed histogram:** Histogram H_d is smoothed by convolving it with a low pass filter of length 2*HFW+1 and defined at index *k* as:

$$F_k = \frac{0.5 + 0.5 * \cos\left[\pi * (k - \mathrm{HFW})/(1 + \mathrm{HFW})\right]}{\sum_{i=0}^{2*\mathrm{HFW}} \left\{0.5 + 0.5 * \cos\left[\pi * (i - \mathrm{HFW})/(1 + \mathrm{HFW})\right]\right\}}, \quad 0 \le k \le 2 * \mathrm{HFW}\ .$$

When considering the smoothed histogram SH_d that results from this step, the HFW bins on each end of SH_d are treated as undefined. This restricts the video delays that can be estimated to plus or minus (UNCERTAINTY-HFW). Smoothing of the histogram increases the robustness of the video delay estimates.

11. **Examine the histogram information:** From the original histogram, H_d, and the smoothed histogram, SH_d, the following three values are determined:
    - max_H_value: The maximum value of H_d.
    - max_SH_offset: The offset *d* that maximizes SH_d.
    - max_SH_value: The maximum value of SH_d (e.g., at *d* = max_SH_offset).

Next, the following two checks are performed:

- *Was U large enough?* Recall that the first and last HFW bins of H_d are missing from SH_d. Examine the values of H_d in these bins. If (H_d > max_H_value * BELOW_WARN), then the temporal registration uncertainty is too small. The algorithm must be re-run with a larger U. The values of *d* to check are (-U ≤ *d* < –U + HFW) and (U – HFW < *d* ≤ U).
- *Does SH_d have one well-defined delay?* Examine SH_d, except within DELTA of max_SH_offset. If (SH_d > max_SH_value * BELOW_WARN) for any video delay *d* where (-U ≤ *d* < max_SH_offset – DELTA) or (max_SH_offset + DELTA < *d* ≤ U), then temporal registration is ambiguous.

If the above two checks pass, then the video delay given by max_SH_offset is chosen as the best average temporal registration for the scene.

6.4.1.5 Observations and Conclusions

The frame-based video delay measurement algorithm uses sub-sampled original and processed video sequences. This algorithm is suitable for aligning video in a fully automated out-of-service environment, prior to performing video quality measurements. The frame-based video delay measurement algorithm

estimates the temporal registration for each image, forms histograms of those individual estimates, and then uses the most commonly indicated delay as the overall video delay -- or temporal registration -- for the selected sequence of video frames.

The delay indicated at the final stage of the algorithm (step 11 of 6.4.1.4) may be different from the delay a viewer might choose, if aligning the scenes by eye. Viewers tend to focus on motion, aligning the high motion parts of the scene, where the frame-based algorithm chooses the most often observed delay over all of the frames that were examined. These overall delay histograms can be examined to determine the extent and statistics of any variable video delay present in the HRC.

6.4.2 Applying Temporal Registration Correction

All of the quality features will require that the temporal delay calculated herein be removed. For positive delays, remove frames from the beginning of the processed file and the end of the original file. For negative delays, remove frames from the end of the processed file and the beginning of the original file. When reframing interlaced video sequences, the processed sequence is reframed. Thus, one field should be removed from the beginning and end of the processed video sequence in addition to the above. Simultaneously, one frame must be removed from either the beginning of the original video file (i.e., -1 field delay overall) or the end of the original video file (i.e., +1 field delay overall).

Correcting for temporal registration will, in effect, shorten the length of available images in the video sequence. For simplicity, all further calculations will be based on the number of video frames available after all calibration corrections have been applied.

7 Quality Features

7.1 Introduction

A quality *feature* is defined as a quantity of information associated with, or extracted from, a spatial-temporal sub-region of a video stream (either original or processed). The feature streams that are produced are a function of space and time. By comparing features extracted from the calibrated processed video with features extracted from the original video, a set of quality *parameters* (clause 8) can be computed that are indicative of perceptual changes in video quality. This section describes a set of quality features that characterize perceptual changes in the spatial, temporal, and chrominance properties of video streams. Normally, a perceptual filter is applied to the video stream to enhance some property of perceived video quality, such as edge information. After this perceptual filtering, features are extracted from spatial-temporal (S-T) sub-regions using a mathematical function (e.g., standard deviation). Finally, a perceptibility threshold is applied to the extracted features.

For the following discussion, an original feature stream will be denoted as $f_o(s, t)$ and the corresponding processed feature stream will be denoted as $f_p(s, t)$, where s and t are indices that denote the spatial and temporal positions, respectively, of the S-T region within the calibrated original and processed video streams. The features will be assigned lettered subscripts as they are discussed in the following sections, where the subscripted letters are chosen to be indicative of what the feature is measuring. All features operate on frames within a calibrated video sequence (see clause 6); any interlace issues are addressed during calibration. All features operate independently of image size (i.e., S-T region size does not change when the image size changes).[9]

In summary, feature calculations perform the following steps. Some features may not require those steps marked [Optional].

1. [Optional] Apply a perceptual filter.

[9] There is an implicit assumption that the viewing distance as a function of picture height remains fixed (e.g., closer viewing distances are used for smaller images). See clause 9 for further comments regarding the assumed viewing distance.

2. Divide the video stream into S-T regions.
3. Extract features, or summary statistics, from each S-T region (e.g., mean, standard deviation).
4. [Optional] Apply a perceptibility threshold.

Some features may utilize two or more different perceptual filters.

7.1.1 S-T Regions

In general, features are extracted from localized S-T regions after the original and processed video streams have been perceptually filtered. The S-T regions are positioned to divide the video streams into abutting S-T regions. Since the processed video has been calibrated, for each processed video S-T region there exists an original S-T region spanning the identical spatial and temporal position within the video stream. Features are extracted from each S-T region by calculating summary statistics or some other mathematical function over the S-T region of interest.

Each S-T region describes a block of pixels. S-T region sizes are described by; (1) the number of pixels horizontally; (2) the number of frame lines vertically; and (3) the time duration of the region, given in units of equivalent video frames referenced to a 30 fps video system.[10] Figure 10 illustrates a S-T region of 8 horizontal pixels x 8 vertical lines x 6 NTSC video frames, for a total of 384 pixels. When applied to 25 fps video (PAL), this same S-T region spans 8 horizontal pixels x 8 vertical lines x 5 video frames, for a total of 320 pixels.

One fifth of a second is a desirable temporal extent, due to the ease of frame rate conversions (i.e., one fifth of a second results in an integer number of video frames for video systems operating at 10 fps, 15 fps, 25 fps, and 30 fps). The general rule for frame rate conversion is to take the length of the S-T region in 30 fps video frames, divide by 30 and multiply by the frame rate of the video system under test. S-T regions that contain one video frame are presumed to always contain one video frame, independent of the frame rate.

The SROI (see clause 3) encompassed by all S-T regions is identical for the original and calibrated processed video sequences. The SROI must lie entirely within the PVR, possibly with a buffer of pixels as required by any convolutional perceptual filter. The horizontal width of the SROI must be evenly divisible by the S-T region's horizontal extent. Likewise, the vertical height of the SROI must be evenly divisible by the S-T region's vertical extent. A user might further constrain the SROI to encompass a region of particular interest, such as the center of the video frame.

Temporally, the original and calibrated processed video sequences are divided into an identical number of S-T regions, beginning with the first frame of temporally aligned video. If the number of valid frames available cannot be evenly divided by the S-T region's temporal extent, frames at the end of the clip are dropped from consideration.

For some features such as those presented in 7.2, the 8x8_6F block achieves close to maximum correlation with subjective ratings. It should be noted, however, that the correlation decreases *slowly* as one moves away from the optimum S-T region size. Horizontal and vertical widths up to 32 or even larger and temporal widths up to 30 frames can be used with satisfactory results, giving the objective measurement system designer considerable flexibility in adapting the features to the available storage or transmission bandwidth [22].

[10] All time durations in this standard will be referenced to the equivalent number of video frames from a 30 fps video system. Thus, time durations of 6 frames (F) is used to represent both 6 frames from an NTSC system (6/30) and 5 frames from a PAL system (5/25). In addition, 30 fps and 29.97 fps are used interchangeably in this standard, as this slight difference in frame rate is inconsequential for computation of VQM.

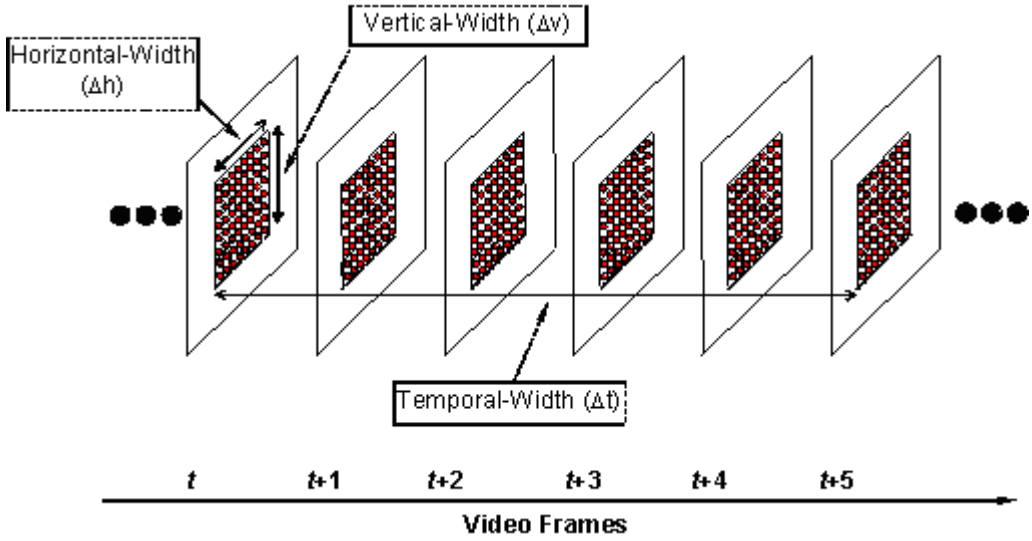

Figure 10 - Example spatial-temporal (S-T) region size for extracting features

After the video stream has been divided into S-T regions, the temporal axis of the feature (*t*) no longer corresponds to individual frames. Rather, the temporal axis contains a number of samples equal to the number of valid frames in the calibrated video sequence divided by the temporal extent of the S-T region.

When computing two or more features simultaneously, further considerations become important. Ideally, all features should be calculated for the same SROI.

7.2 Features Based on Spatial Gradients

Features derived from spatial gradients can be used to characterize perceptual distortions of edges. For example, a general loss of edge information results from blurring while an excess of horizontal and vertical edge information can result from block distortion or tiling. The Y components of the original and processed video streams are filtered using horizontal and vertical edge enhancement filters. Next, these filtered video streams are divided into spatial-temporal (S-T) regions from which features, or summary statistics, are extracted that quantify the spatial activity as a function of angular orientation. Then, these features are clipped at the lower end to emulate perceptibility thresholds. The edge enhancement filters, the S-T region size, and the perceptibility thresholds were selected based on Rec. 601 video that has been subjectively evaluated at a viewing distance of six picture heights. Figure 11 presents an overview of the algorithm used to extract features based on spatial gradients.

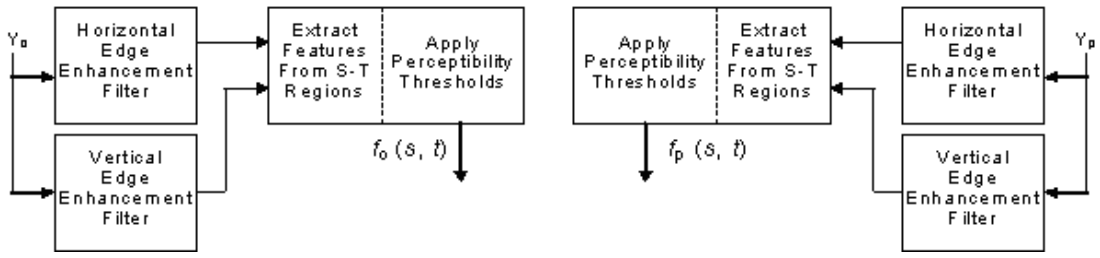

Figure 11 - Overview of algorithm used to extract spatial gradient features

7.2.1 Edge Enhancement Filters

The original and processed Y (luminance) video *frames* are first processed with horizontal and vertical edge enhancement filters that enhance edges while reducing noise. The two filters shown in Figure 12 are applied separately, one to enhance horizontal pixel differences while smoothing vertically (left filter), and the other to enhance vertical pixel differences while smoothing horizontally (right filter).

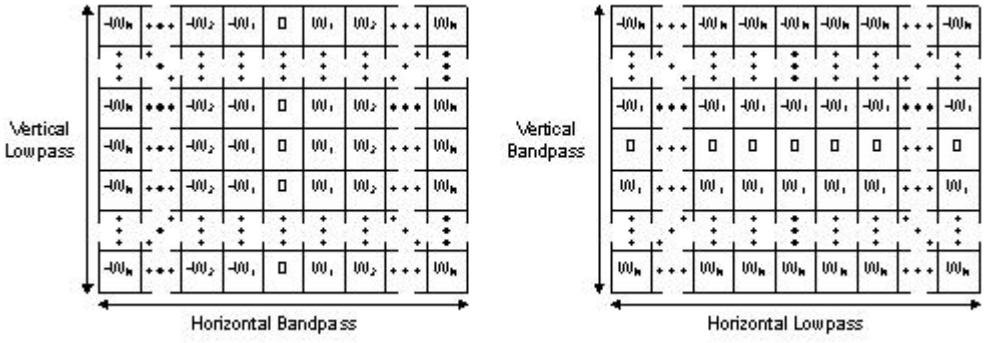

Figure 12 - Edge enhancement filters

The two filters are transposes of each other, have size 13 x 13, and have filter weights given by:

$$w_x = k * \left(\frac{x}{c}\right) * \exp\left\{-\left(\frac{1}{2}\right)\left(\frac{x}{c}\right)^2\right\},$$

where x is the pixel displacement from the center of the filter (0, 1, 2, …, N), c is a constant that sets the width of the bandpass filter, and k is a normalization constant selected such that each filter would produce the same gain as a true Sobel filter [16]. The optimal amount of horizontal bandpass filtering for a viewing distance of six times picture height was found to be given by the c = 2 filter, which has a peak response at about 4.5 cycles/degree. The bandpass filter weights that were used are given by:

[-.0052625, -.0173446, -.0427401, -.0768961, -.0957739, -.0696751, 0, .0696751, .0957739, .0768961, .0427401, .0173446, .0052625].

Note that the filters in Figure 12 have a flat low-pass response. A flat low-pass response produced the best quality estimate and has the added advantage of being computationally efficient (e.g., for the left filter in Figure 12, one merely has to sum the pixels in a column and multiply once by the weight).

7.2.2 Description of Features f_{SI13} and f_{HV13}

This section describes the extraction of two spatial activity features from S-T regions of the edge-enhanced original and processed video streams from 7.2.1. These features will be used to detect spatial impairments such as blurring and blocking. The filter shown in Figure 12 (left) enhances spatial gradients in the horizontal (H) direction while the transpose of this filter (right) enhances spatial gradients in the vertical (V) direction. The response at each pixel from the H and V filters can be plotted on a two dimensional diagram -- such as the one shown in Figure 13 -- with the H filter response forming the abscissa value and the V filter response forming the ordinate value. For a given image pixel located at row i, column j, and time t, the H and V filter responses will be denoted as $H(i, j, t)$ and $V(i, j, t)$, respectively. These responses can be converted into polar coordinates (R, θ) using the relationships:

$R(i,j,t) = \sqrt{H(i,j,t)^2 + V(i,j,t)^2}$, and

$$\theta(i,j,t) = \tan^{-1}\left[\frac{V(i,j,t)}{H(i,j,t)}\right].$$

The first feature is a measure of overall spatial information (SI) and hence is denoted as f_{SI13} since images were pre-processed using the 13 x 13 filter masks shown in Figure 12. This feature is computed simply as the standard deviation (*std*) over the S-T region of the *R*(*i*, *j*, *t*) samples, and then clipped at the perceptibility threshold of *P* (i.e., if the result of the *std* calculation falls below *P*, f_{SI13} is set equal to *P*), namely

$$f_{SI13} = \{std[R(i,j,t)]\}|_P : \; i,j,t \in \{\text{S} - \text{T Region}\}.$$

This feature is sensitive to changes in the overall amount of spatial activity within a given S-T region. For instance, localized blurring produces a reduction in the amount of spatial activity, whereas noise produces an increase. The recommended threshold *P* for this feature is 12.

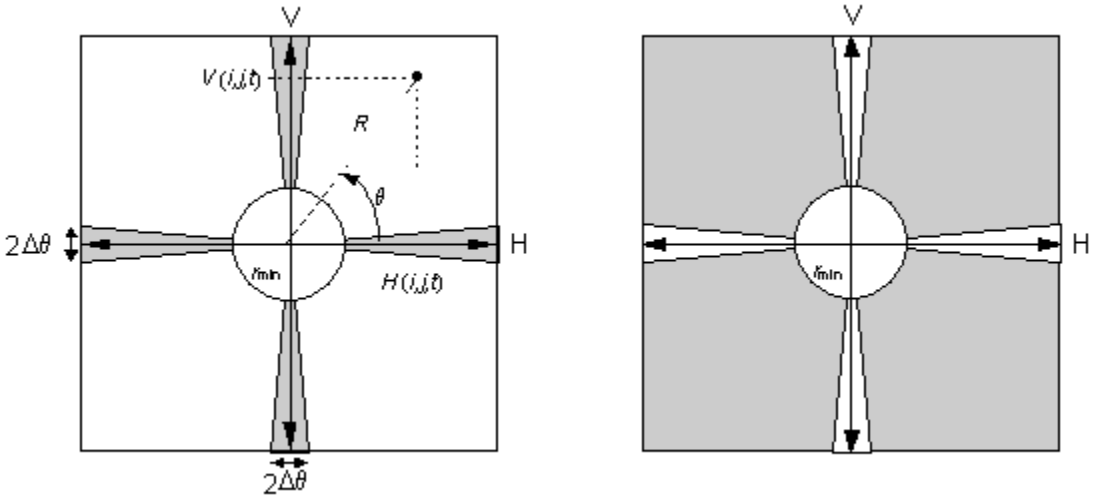

Figure 13 - Division of horizontal (H) and vertical (V) spatial activity into HV (left) and (right) distributions

The second feature, f_{HV13}, is sensitive to changes in the angular distribution, or orientation, of spatial activity. Complementary images are computed with the shaded spatial gradient distributions shown in Figure 13. The image with horizontal and vertical gradients, denoted as *HV*, contains the *R*(*i*, *j*, *t*) pixels that are horizontal or vertical edges (pixels that are diagonal edges are zeroed). The image with the diagonal gradients, denoted as $\overline{HV}$ **,** contains the *R*(*i*, *j*, *t*) pixels that are diagonal edges (pixels that are horizontal or vertical edges are zeroed). Gradient magnitudes *R*(*i*, *j*, *t*) less than r_{min} are zeroed in both images to assure accurate θ computations. Pixels in *HV* and $\overline{HV}$ can be represented mathematically as:

$$HV(i,j,t)=\begin{cases} R(i,j,t) \quad if \;\; R(i,j,t)\geq r_{\min} \;\; and \;\; m\frac{\pi}{2}-\Delta\theta < \theta(i,j,t) < m\frac{\pi}{2}+\Delta\theta \quad (m=0,1,2,3) \\ 0 \quad otherwise \end{cases},$$

and:

$$\overline{HV}(i,j,t)=\begin{cases} R(i,j,t) \quad if \;\; R(i,j,t)\geq r_{\min} \;\; and \;\; m\frac{\pi}{2}+\Delta\theta \leq \theta(i,j,t) \leq (m+1)\frac{\pi}{2}-\Delta\theta \quad (m=0,1,2,3) \\ 0 \quad otherwise \end{cases}$$

where:

$$i,j,t\in\{\text{S}-\text{T Region}\}.$$

For the computation of *HV* and $\overline{HV}$ above, the recommended value for r_{min} is 20 and the recommended value for $\Delta\theta$ is 0.225 radians. Feature f_{HV13} for one S-T region is then given by the ratio of the mean of *HV* to the mean of $\overline{HV}$, where these resultant means are clipped at their perceptibility thresholds *P*, namely:

$$f_{HV13}=\{mean[HV(i,j,t)]\}\big|_P \Big/ \{mean[\overline{HV}(i,j,t)]\}\big|_P .$$

The recommended perceptibility threshold *P* for the mean of *HV* and $\overline{HV}$ is 3. The f_{HV13} feature is sensitive to changes in the angular distribution of spatial activity within a given S-T region. For example, if horizontal and vertical edges suffer more blurring than diagonal edges, f_{HV13} of the processed video will be less than f_{HV13} of the original video. On the other hand, if erroneous horizontal or vertical edges are introduced, say in the form of blocking or tiling distortions, then f_{HV13} of the processed video will be greater than f_{HV13} of the original video. The f_{HV13} feature thus provides a simple means to include variations in the sensitivity of the human visual system with respect to angular orientation.[11]

7.3 Features Based on Chrominance Information

This section presents a single feature that can be used to measure distortions in the chrominance signals (C_B, C_R). For a given image pixel located at row *i*, column *j*, and time *t*, let $C_B(i, j, t)$ and $C_R(i, j, t)$ represent the Rec. 601 C_B and C_R values.[12] The components of a two-dimensional chrominance feature vector, $\underline{f}_{COHER\_COLOR}$, are computed as the mean (*mean*) over the S-T region of the $C_B(i, j, t)$ and $C_R(i, j, t)$ samples, respectively, giving more perceptual weight to the C_R component:

$$\underline{f}_{COHER\_COLOR}=\left(mean[C_B(i,j,t)], W_R * mean[C_R(i,j,t)]\right): \; i,j,t\in\{\text{S}-\text{T Region}\}, \text{ and } W_R=1.5 .$$

The above equation performs coherent integration (hence the name $\underline{f}_{COHER\_COLOR}$) since the phase relationship between C_B and C_R is preserved. If one is familiar with a vectorscope, the value of the

[11] This discussion of f_{HV13}, though true in general, is somewhat simplified. For instance, when encountering some shapes the f_{HV13} filter behaves in a manner that may be counter-intuitive (e.g., a corner formed by the joining of a vertical and horizontal line will result in diagonal energy).

[12] Gain and offset corrections are not applied to the C_B and C_R image planes. See 6.3.3.

chrominance feature when examining color bar signals is readily apparent. For general-purpose scenes, one can visualize the chrominance feature vector's usefulness for measuring distortions in chrominance for blocks of video that span a range of spatial and temporal extent. However, if S-T region size is too large, then many colors could be included in the calculation, and the usefulness of f_{COHER_COLOR} is reduced. An S-T region size of 8 horizontal pixels x 8 vertical lines x (1 to 3) video frames produces a robust chrominance feature vector (actually 4 horizontal C_B and C_R pixels, since these signals are sub-sampled by two in the horizontal direction for Rec. 601 sampling).

7.4 Features Based on Contrast Information

Features that measure localized contrast information are sensitive to quality degradations such as blurring (e.g., contrast loss) and added noise (e.g., contrast gain). One localized contrast feature, f_{CONT}, is easily computed for each S-T region from the Y luminance image as:

$$f_{CONT} = \left\{ std\left[Y(i,j,t) \right] \right\}\Big|_P : i,j,t \in \left\{ \text{S} - \text{T Region} \right\}.$$

The recommended perceptibility threshold P for the f_{CONT} feature is between four and six.

7.5 Features Based on Absolute Temporal Information (ATI)

Features that measure distortions in the flow of motion are sensitive to quality degradations such as dropped or repeated frames (motion loss) and added noise (motion gain). An absolute temporal information feature, f_{ATI}, is computed for each S-T region by first generating a motion video stream that is the absolute value of the difference between consecutive video frames at time t and t-1, and then computing the standard deviation over the S-T region. Mathematically, this process will be represented as:

$$f_{ATI} = \left\{ std\ \left| Y(i,j,t) - Y(i,j,t-1) \right| \right\}\Big|_P : i,j,t \in \left\{ \text{S} - \text{T Region} \right\}.$$

The recommended perceptibility threshold P for the f_{ATI} feature is between one and three.

The use of a previous frame introduces considerations beyond those required by the other features. When calculating f_{ATI} jointly with another feature (e.g., $f_{CONTRAST\_ATI}$ from 7.6) or for use in a model (see clause 9), the requirement of an extra frame complicates the task of placement of S-T regions (see 7.1.1).

7.6 Features Based on the Cross Product of Contrast and Absolute Temporal Information

The perceptibility of spatial impairments can be influenced by the amount of motion that is present. Likewise, the perceptibility of temporal impairments can be influenced by the amount of spatial detail that is present. A feature derived from the cross product of contrast information and absolute temporal information can be used to partially account for these interactions. This feature, denoted as $f_{CONTRAST\_ATI}$, is computed as the product of the features in 7.4 and 7.5.[13] The recommended perceptibility threshold P = 3 is applied separately to each feature (f_{CONT} and f_{ATI}) before computing their cross product. Impairments will be more visible in S-T regions that have a low cross product than in S-T regions that have a high cross product. This is particularly true of impairments like noise and error blocks [2].

The requirement of an extra frame for f_{ATI} complicates $f_{CONTRAST\_ATI}$ slightly, since the S-T regions used by both f_{CONT} and f_{ATI} must be placed identically. Either one frame at the beginning of the video sequence must be left unused for f_{ATI}, or the S-T regions located at the beginning of the video sequence must

[13] A standard cross product of the f_{CONT} and f_{ATI} features (i.e., $f_{CONT} * f_{ATI}$) is used for the processed $f_p(s, t)$ and original $f_o(s, t)$ features in the *ratio_loss* and *ratio_gain* comparison functions described in 8.2.1. However, for the *log_loss* and *log_gain* comparison functions, the processed and original features are computed as $\log_{10}[f_{CONT}] * \log_{10}[f_{ATI}]$, and the comparison functions use subtraction (i.e., $f_p(s, t) - f_o(s, t)$ rather than $\log_{10}[f_p(s, t) / f_o(s, t)]$).

contain one fewer frame (e.g., given a temporal extent of 6F, the first f_{ATI} S-T region would use 5F instead of 6F). The parameters and models specified herein presume the second solution will be used.

8 Quality Parameters

8.1 Introduction

Quality parameters that measure distortions in video quality due to gains and losses in the feature values are first calculated for each S-T region by comparing the original feature values, $f_o(s, t)$, with the corresponding processed feature values, $f_p(s, t)$ (see 8.2). Several functional relationships are used to emulate the visual masking of impairments for each S-T region. Next, error-pooling functions across space and time emulate how humans deduce subjective quality ratings. Error pooling across space will be referred to as spatial collapsing (see 8.3), and error pooling across time will be referred to as temporal collapsing (see 8.4). Sequential application of the spatial and temporal collapsing functions to the stream of S-T quality parameters produces quality parameters for the entire video clip, which is nominally 5 to 10 seconds in duration. The final time-collapsed parameter values may be scaled and clipped (see 8.5) to account for nonlinear relationships between the parameter value and perceived quality and to further reduce the parameter's sensitivity.

In summary, parameter calculations perform the following steps. Some features may not require the [Optional] step.

1. Compare original feature values with processed feature values.
2. Perform spatial collapsing.
3. Perform temporal collapsing.
4. [Optional] Perform nonlinear scaling and/or clipping.

All parameters are designed to be either all positive or all negative. A parameter value of zero indicates no impairment.

8.2 Comparison Functions

The perceptual impairment at each S-T region is calculated using functions that model visual masking of the spatial and temporal impairments. This section presents the masking functions that are used by the various parameters to produce quality parameters as a function of space and time.

8.2.1 Error Ratio and Logarithmic Ratio

Loss and gain are normally examined separately, since they produce fundamentally different effects on quality perception (e.g., loss of spatial activity due to blurring and gain of spatial activity due to noise or blocking). Of the many comparison functions that have been evaluated, two forms have consistently produced the best correlation to subjective ratings. Each of these forms can be used with either gain or loss calculations for a total of four basic S-T comparison functions. The four primary forms are:

1. $$ratio\_loss(s,t) = np\left\{\frac{f_p(s,t) - f_o(s,t)}{f_o(s,t)}\right\},$$

2. $$ratio\_gain(s,t) = pp\left\{\frac{f_p(s,t) - f_o(s,t)}{f_o(s,t)}\right\},$$

3. $\log\_loss(s,t) = np\left\{\log_{10}\left[\frac{f_p(s,t)}{f_o(s,t)}\right]\right\}$, and

4. $\log\_gain(s,t) = pp\left\{\log_{10}\left[\frac{f_p(s,t)}{f_o(s,t)}\right]\right\}$,

where *pp* is the positive part operator (i.e., negative values are replaced with zero), and *np* is the negative part operator (i.e., positive values are replaced with zero).

These visual masking functions imply that impairment perception is inversely proportional to the amount of localized spatial or temporal activity that is present. In other words, spatial impairments become less visible as the spatial activity increases (i.e., spatial masking), and temporal impairments become less visible as the temporal activity increases (i.e., temporal masking). While the logarithmic and ratio comparison functions behave very similarly, the logarithmic function tends to be slightly more advantageous for gains, while the ratio function tends to be slightly more advantageous for losses. The logarithm function has a larger dynamic range, and this is useful when the processed feature values greatly exceed the original feature values.

8.2.2 Euclidean Distance

Another useful S-T comparison function is simple Euclidean distance, represented by the length of the difference vector between the original feature vector $f_o(s, t)$ and the corresponding processed feature vector, $f_p(s, t)$:

$$euclid(s,t) = \left\| \underline{f}_p(s,t) - \underline{f}_o(s,t) \right\|.$$

Figure 14 gives an illustration of Euclidean distance for a two-dimensional feature vector extracted from a S-T region (e.g., the $\underline{f}_{COHER\_COLOR}$ feature vector of clause 7.3), where *s* and *t* are indices that denote the spatial and temporal positions, respectively, of the S-T region within the calibrated original and processed video streams. The dashed line in Figure 14 shows the Euclidean distance. The Euclidean distance measure can be generalized for feature vectors that have an arbitrary number of dimensions.

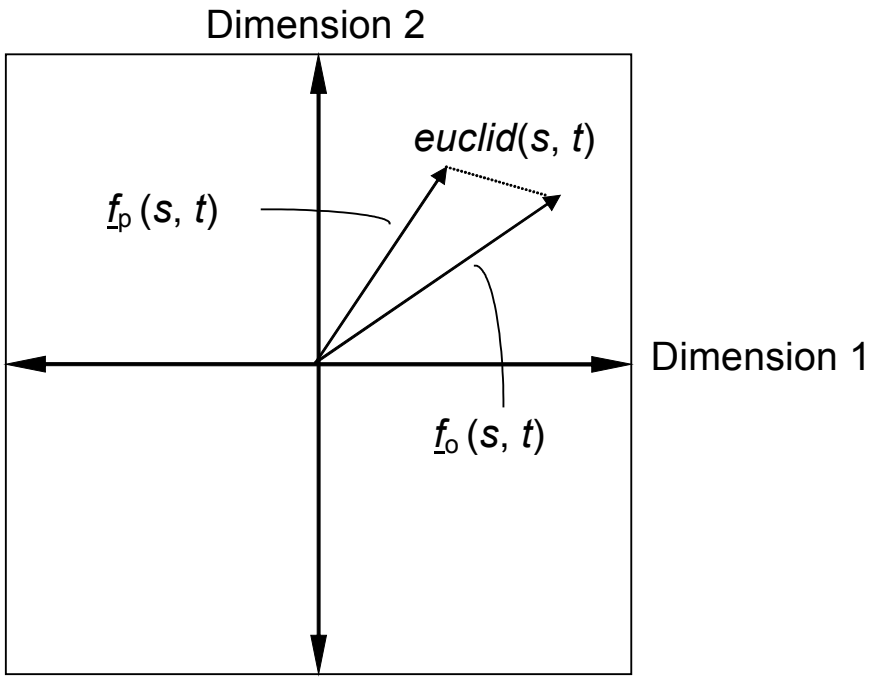

Figure 14 - Illustration of the Euclidean distance euclid(s, t) for a two-dimensional feature

8.3 Spatial Collapsing Functions

The parameters from the S-T regions (from 8.2) form three-dimensional matrixes spanning one temporal axis and two spatial dimensions (i.e., horizontal and vertical placement of the S-T region). Next, impairments from the S-T regions with the same time index *t* are pooled using a spatial collapsing function. Spatial collapsing yields a time history of parameter values. This time history of parameter values, denoted generically as *p*(*t*), must then be temporally collapsed using a temporal collapsing function given in 8.4. Table 1 presents a summary of the most commonly used spatial collapsing functions.

Extensive investigation has revealed that the optimal spatial collapsing functions normally involve some form of worst case processing, like the average of the worst 5% of the distortions observed over the spatial index *s* ([20]-[23]). This is because localized impairments tend to draw the focus of the viewer, making the worst part of the picture the predominant factor in the subjective quality decision. For example, the spatial collapsing function "*above95%*" is computed at each temporal index *t* for the *log_gain*(*s,t*) function in 8.2.1 as the average of the most positive 5% of the values over the spatial index *s*.[14] This amounts to sorting the gain distortions from low to high at each temporal index *t* and averaging those distortions that are above the 95% threshold (since more positive values imply greater distortion). Similarly, loss distortions such as those produced by the *ratio_loss*(*s,t*) function in 8.2.1 would be sorted at each temporal index *t*, but the average of those distortions that are "*below5%*" is used (since losses are negative).

8.4 Temporal Collapsing Functions

The parameter time history results *p*(*t*) output from the spatial collapsing function (from 8.3) are next pooled using a temporal collapsing function to produce an objective parameter *p* for the video clip, which

[14] Notice that the time index, *t*, does not indicate individual frames (see 7.1.1) here. Instead, each value of *t* corresponds to those S-T regions having the same time extent.

is nominally 4 to 10 seconds in length. Viewers seem to use several temporal collapsing functions when subjectively rating video clips that are approximately 10 seconds in length. The *mean* over time is indicative of the average quality that is observed during the time period. The *90%* and *10%* levels over time are indicative of the worst transient quality that is observed for gains and losses, respectively (e.g., digital transmission errors may cause a 1 to 2 second disturbance in the processed video). After temporal collapsing, a given parameter *p* is either all negative or all positive, but not both. Talbe 2 presents a summary of the most commonly used temporal collapsing functions.

Table 1 - Spatial Collapsing Functions and Their Definitions

Spatial Collapsing Function	Definition
below5%	For each temporal index t, sort the parameter values from low to high. Compute the average of all the parameter values that are less than or equal to the 5% threshold level. For loss parameters, this spatial collapsing function produces a parameter that is indicative of the worst quality over space.
above95%	For each temporal index t, sort the parameter values from low to high. Compute the average of all the parameter values that are greater than or equal to the 95% threshold level. For gain parameters, this spatial collapsing function produces a parameter that is indicative of the worst quality over space.
mean	For each temporal index t, compute the average of all the parameter values. This spatial collapsing function produces a parameter that is indicative of the average quality over space.
std	For each temporal index t, compute the standard deviation of all the parameter values. This spatial collapsing function produces a parameter that is indicative of the quality variations over space.
below5%tail	For each temporal index t, sort the parameter values from low to high. Compute the average of all the parameter values that are less than or equal to the 5% threshold level, and then subtract the 5% level from this average. For loss parameters, this spatial collapsing function allows one to measure the spread of the worst quality levels over space. It is useful for measuring the perceptual quality effects of spatially localized distortions.
above99%tail	For each temporal index t, sort the parameter values from low to high. Compute the average of all the parameter values that are greater than or equal to the 99% threshold level, and then subtract the 99% level from this average. For gain parameters, this spatial collapsing function allows one to measure the spread of the worst quality levels over space. It is useful for measuring the perceptual quality effects of spatially localized distortions.

Table 2 - Temporal Collapsing Functions and Their Definitions

Temporal Collapsing Function	Definition
10%	Sort the time history of the parameter values from low to high and select the 10% threshold level. For loss parameters, this temporal collapsing function produces a parameter that is indicative of the worst quality over time. For gain parameters, it produces a parameter that is indicative of the best quality over time.
25%	Sort the time history of the parameter values from low to high and select the 25% threshold level.
50%	Sort the time history of the parameter values from low to high and select the 50% threshold level.
90%	Sort the time history of the parameter values from low to high and select the 90% threshold level. For loss parameters, this temporal collapsing function produces a parameter that is indicative of the best quality over time. For gain parameters, it produces a parameter that is indicative of the worst quality over time.
mean	Compute the mean of the time history of the parameter values. This produces a parameter that is indicative of the average quality over time.
std	Compute the standard deviation of the time history of the parameter values. This temporal collapsing function produces a parameter that is indicative of the quality variations over time.
above90%tail	Sort the time history of the parameter values from low to high and compute the average of all the parameter values that are greater than or equal to the 90% threshold level, and then subtract the 90% level from this average. For gain parameters, this temporal collapsing function allows one to measure the spread of the worst quality levels over time. It is useful for measuring the perceptual quality effects of temporally localized distortions.

8.5 Nonlinear Scaling and Clipping

The all-positive or all-negative temporally collapsed parameter *p* from 8.4 may be scaled to account for nonlinear relationships between the parameter value and perceived quality. It is preferable to remove any nonlinear relationships before building the video quality models (clause 9), since a linear least-squares algorithm will be used to determine the optimal parameter weights. The two nonlinear scaling functions that might be applied are the square root function, denoted by *sqrt*, and the square function, denoted by *square*. If the *sqrt* function is applied to an all-negative parameter, the parameter is first made all positive (i.e., absolute value taken).

Finally, a clipping function denoted as *clip_T*, where *T* is the clipping threshold, might be applied to reduce the sensitivity of the parameter to small impairments. The clipping function replaces any parameter value between the clipping level and zero with the clipping level, and then the clipping level is subtracted from all resulting parameter values. This is represented mathematically as;

$$clip\_T(p) = \begin{cases} \max(p,T) - T & if\ \ p\ is\ all\ \ positive \\ \min(p,T) - T & if\ \ p\ is\ all\ negative \end{cases}.$$

8.6 Parameter Naming Convention

This clause summarizes the technical naming convention used for video quality parameters. This convention assigns to each parameter a lengthy name consisting of identifying words (sub-names) separated by underscores. The technical parameter name summarizes the exact process used to calculate the parameter. Each sub-name identifies one function or step in the process of calculating the parameter. Sub-names are listed in the order in which they occur, from left to right. Table 3 summarizes the sub-names used to create the technical parameter name, listed in the order that they occur. Clause 8.6.1 provides examples of technical parameter names and their associated sub-names fromTable 3.

Table 3 - Technical Naming Convention Used for Video Quality Parameters

Sub-name	Definition	Examples
Color	The color space image planes used by the parameter.	*Y* for luminance image plane *color* for (C_B, C_R) image planes
Feature Specific	The "Feature Specific" sub-name describes the calculations that make this parameter unique. All other sub-names that follow are generic processes that can be used by many different types of parameters. The "Feature Specific" sub-name is usually the name of the feature that is extracted from the "Color" plane at this point in the flow, hence the location of this sub-name. However, information not otherwise covered by the naming convention can also be included here. For example, the HV parameter applies the "Block Statistic" sub-name separately to the *HV* and $\overline{HV}$ image planes. The subsequent ratio of *HV* to $\overline{HV}$ is specified by the "Feature Specific" sub-name (i.e., rather than occupying a separate sub-name after the "Block Statistic").	*si13* for the f_{SI13} feature in 7.2.2 *hv13_angleX.XXX_rminYY* for the f_{HV13} feature in 7.2.2, where *X.XXX* is $\Delta\theta$ and *YY* is the r_{min} *coher_color* for the f_{COHER_COLOR} feature in 7.3 *cont* for the f_{CONT} feature in 7.4 *ati* for the f_{ATI} feature in 7.5 *contrast_ati* for the $f_{CONTRAST\_ATI}$ feature in 7.6
Block Shift	Present when S-T blocks slide (e.g., overlap in time). When absent, blocks are assumed to abut in time.	*sliding*
Full Image	Present when the S-T block size contains the entire valid region of the image. When absent, the "Block Size" sub-name must be present.	*image*
Block Size	Present when the image is divided into S-T blocks (see 7.1.1). For consistency, block size is always indicated relative to the luminance (Y) plane's frame lines and frame pixels. Thus, for 4:2:2 sampled video, color blocks will actually contain half the specified number of pixels horizontally. When absent, the "Full Image" sub-name must be present.	*8x8* for blocks that include 8 frame lines vertically by 8 frame pixels horizontally *128x128* for blocks that include 128 frame lines vertically by 128 frame pixels horizontally
Block Frames	Indicates the temporal extent of the S-T blocks (see 7.1.1), referenced to 30 frames per second (fps) video. For example, *6F* is used to represent one fifth of a second, regardless of the frame rate of the video being measured (e.g., 5 frames from a 25 fps system, 3 frames from a 15 fps system, 2 frames from a 10 fps system).	*1F* for a temporal extent of one frame *6F* for a temporal extent of one fifth of a second
Block Statistic	The statistical function used to extract the feature from each S-T region, producing one number for each S-T block of pixels. Present unless "Block Size" = *1x1* (i.e., 1 pixel). Before the Block Statistic has been applied, intermediate results contain time histories of images with one number per pixel (i.e., filtered images); afterward, intermediate results contain one number per each S-T region (i.e., feature images). Parameters that have two image planes (e.g., *hv13* and *coher_color*) will apply the Block Statistic separately to both image planes, producing two feature images.	*mean* is the average of the pixel values *std* is the standard deviation of the pixel values *rms* is the root mean square of the pixel values
Perceptibility Threshold	The values produced by the "Block Statistic" may be clipped at a perceptibility threshold *P*. Values between zero and this threshold are replaced with the threshold.	*3* for a minimum feature value of 3.0 *12* for a minimum feature value of 12.0
Comparison Function	The function used to compare features extracted from the original and processed feature streams (see 8.2). Before the Comparison Function, the intermediate results contain time histories of original and processed feature images; afterward the intermediate results contain a time history of parameter images.	*log_gain* (see 8.2.1) *ratio_loss* (see 8.2.1) *euclid* (see 8.2.2).
Spatial Collapsing Function	See 8.3. The function is applied to each parameter image (e.g., all S-T regions having the same temporal index) and produces a time history of parameter values. Before spatial collapsing, intermediate results consist of parameter images containing one value for each S-T block; afterward, intermediate results are a time history of numbers (i.e., parameter time history). Must be present for all parameters except "Full Image" parameters.	See Table 1

Sub-name	Definition	Examples
Temporal Collapsing Function	See 8.4. The function is applied to the parameter time history and produces one parameter value for the entire video sequence. After temporal collapsing, the parameter contains either all negative values or all positive values, but not both. Zero is associated with no impairment, and parameter values further from zero have higher impairments. Must be present for all parameters.	See Table 2
Nonlinear Function	See 8.5. Examination of the parameter's values may indicate that the parameter should be scaled in a nonlinear fashion to linearly track the subjective data. The Nonlinear Function performs this final scaling. If the *sqrt* function is applied to an all-negative parameter, the parameter is first made all positive (i.e., absolute value taken).	*sqrt* for the square root of the temporally collapsed parameter value *square* for the square of the temporally collapsed parameter value
Clipping Function	See 8.5. Final examination of the parameter values may indicate a need to further reduce the sensitivity of the parameter to small impairments (e.g., parameter values near zero). Replace any value between the clipping level *T* and zero with the clipping level, and then subtract the clipping level from all resulting parameter values.	*clip_0.45* If parameter values are positive, replace all values less than 0.45 with 0.45 and then subtract 0.45 from all the parameter values. If parameter values are negative, replace all values greater than –0.45 with –0.45 and then add 0.45 to all the parameter values.

8.6.1 Example Parameter Names

This section includes five example technical names, and a step-by-step description of the sub-naming procedure given in Table 3.

Y_si13_8x8_6F_std_6_ratio_loss_below5%_mean

Y means that the luminance image plane is used. *si13* represents filtering of those images with the 13x13 spatial masks in 7.2.1 in preparation for extraction of the f_{SI13} feature in 7.2.2. *8x8_6F* represents dividing the video stream into S-T regions containing eight frame lines vertically by eight pixels horizontally by one fifth of a second temporally (i.e., 6 NTSC frames, 5 PAL frames). *std* represents taking the standard deviation of each block. *6* represents application of a perceptibility threshold, replacing all standard deviation values below 6.0 with a value of 6.0. *ratio_loss* represents comparing the original and processed features from each block using the *ratio_loss* function. *below5%* represents spatially collapsing the parameter values at each time index using the *below5%* function. *mean* represents temporally collapsing the parameter time history using the *mean* function.

color_coher_color_8x8_1F_mean_euclid_std_10%_clip_0.8

color represents using the C_B and C_R image planes. *coher_color* represents preservation of the phase relationship between the C_B and C_R images (by treating them separately) in preparation for extraction of the f_{COHER_COLOR} feature in 7.3. *8x8_1F* represents dividing each frame into blocks that are 8 frame lines high by 4 C_B and C_R pixels wide (due to 4:2:2 subsampling of the C_B and C_R image planes) by 1 frame in time. *mean* represents taking the mean value of each block. *euclid* represents computing the Euclidean distance between original vectors (C_B, C_R) and processed vectors (C_B, C_R) for each S-T block. *std* represents the *std* spatial collapsing function. *10%* represents the *10%* temporal collapsing function. *clip_0.8* represents clipping the final parameter value at a minimum of 0.8 (i.e., replacing all values below 0.8 with 0.8, and then subtracting 0.8).

Y_hv13_angle0.225_rmin20_8x8_6F_mean_3_ratio_loss_below5%_mean_square_clip_0.05

Y means that the luminance image plane is used. *hv13* represents filtering of the Y images with the 13x13 spatial masks in 7.2.1 in preparation for extraction of the f_{HV13} feature in 7.2.2 (i.e., the *HV* and

$\overline{HV}$ images are created and treated separately until after the Perceptibility Threshold). *angle0.225* and *rmin20* represents a $\Delta\theta$ of 0.225 radians and an r_{min} of 20 for calculation of the f_{HV13} feature. *8x8_6F* represents dividing the video stream into S-T regions containing eight frame lines vertically by eight pixels horizontally by one-fifth of a second temporally (i.e., 6 NTSC frames, 5 PAL frames). *mean* represents taking the mean value of each S-T block for *HV* and $\overline{HV}$. *3* represents the application of a perceptibility threshold to these means, replacing all values less than 3.0 with 3.0. Next, the f_{HV13} feature in 7.2.2 is calculated as the ratio of clipped means of *HV* to the clipped means of $\overline{HV}$, as specified in *hv13_angle0.225_rmin20*, the Feature Specific sub-name. *ratio_loss* represents using the *ratio_loss* comparison function for each original and corresponding processed f_{HV13} feature extracted from a S-T block. *below5%* specifies the spatial collapsing function. *mean* specifies the temporal collapsing function. *square* specifies the nonlinear function for each time-collapsed parameter value. *clip_0.05* represents the clipping function, where all values below 0.05 are replaced with 0.05, and then 0.05 is subtracted from the result (recall that the all-negative parameter will become an all-positive parameter due to the nonlinear function, *square*).

Y_contrast_ati_4x4_6F_std_3_ratio_gain_mean_10%

Y means the luminance plane is used. *contrast_ati* represents computing two separate filtered versions of the image in preparation for extraction of the $f_{CONTRAST\_ATI}$ feature in 7.6. The first filter, *contrast*, will consider the luminance planes directly (see 7.4). The second filter, *ati*, will consider images generated by taking differences between successive luminance planes (see 7.5). The *contrast* and *ati* images are treated separately until after the Thresholding. *4x4_6F* means that the two video streams are divided into S-T regions containing four frame lines vertically by four pixels horizontally by one-fifth of a second temporally (e.g, 6 NTSC frames, 5 PAL frames). The first S-T block of *ati* images will actually contain only 5 images rather than 6 since an *ati* image cannot be generated for the first frame in the sequence (i.e., there is no earlier image in time available). This exception is specified as part of the Feature Specific sub-name. *std* represents taking the standard deviation of each block. Then, as specified in the Feature Specific sub-name in 7.6, apply a perceptibility threshold of 3 to both the *contrast* and *ati* features (replace all values less than 3 with 3.0). Next, multiply the *contrast* block-value with the *ati* block-value for each S-T block (see footnote in 7.6 for special instructions on how to perform this multiplication) and continue calculations with this combined feature image. *ratio_gain* is the comparison function used to compare each original and processed feature from the S-T blocks. *mean* is the spatial collapsing function. *10%* is the temporal collapsing function.

9 General Model

This section provides a full description of the general model VQM (denoted as VQM_G). The general model is optimized to achieve maximum objective to subjective correlation using a wide range of video quality and bit rates. The general model has objective parameters for measuring the perceptual effects of a wide range of impairments such as blurring, block distortion, jerky/unnatural motion, noise (in both the luminance and chrominance channels), and error blocks (e.g., what might typically be seen when digital transmission errors are present). This model consists of a linear combination of video quality parameters whose naming conventions are described in 8.6. The selection of video quality parameters was determined by the optimization criteria given above. The general model produces output values that range from zero (no perceived impairment) to approximately one (maximum perceived impairment). To place results on the double stimulus continuous quality scale (DSCQS), multiply VQM_G by 100.

The general model was designed based on Rec. 601 video that has been subjectively evaluated at a viewing distance of six picture heights. When analyzing video sequences for different viewing distances, a scaling factor must be applied to the results. As viewing distance increases, impairments become less visible; as viewing distance decreases, impairments become more visible. Care should be taken when comparing results for video sequences that will be viewed at different viewing distances.

VQM_G consists of a linear combination of seven parameters. Four parameters are based on features extracted from spatial gradients of the Y luminance component (see 7.2.2), two parameters are based on features extracted from the vector formed by the two chrominance components (C_B, C_R see 7.3), and one parameter is based on contrast and absolute temporal information features, both extracted from the Y luminance component (see 7.4 and 7.5, respectively). VQM_G is given by:

VQM_G =

{-0.2097 * Y_si13_8x8_6F_std_12_ratio_loss_below5%_10%
+0.5969 * Y_hv13_angle0.225_rmin20_8x8_6F_mean_3_ratio_loss_below5%_mean_square_clip_0.06
+0.2483 * Y_hv13_angle0.225_rmin20_8x8_6F_mean_3_log_gain_above95%_mean
+0.0192 * color_coher_color_8x8_1F_mean_euclid_std_10%_clip_0.6
-2.3416 * [Y_si13_8x8_6F_std_8_log_gain_mean_mean_clip_0.004 | $^{0.14}$]
+0.0431 * Y_contrast_ati_4x4_6F_std_3_ratio_gain_mean_10%
+0.0076 * color_coher_color_8x8_1F_mean_euclid_above99%tail_std} | $_{0.0}$

Remember, that the above features for the general model with a “6F” time extent will actually contain five PAL (625-line) video frames.

The square on the hv_loss parameter is necessary to linearize the parameter response with respect to the subjective data. Note that since the hv_loss parameter becomes positive after the square, a positive multiplying weight is used. Also note that the hv_loss parameter is clipped at 0.06, the color parameter is clipped at 0.6, and the si_gain parameter is clipped at 0.004. The si_gain parameter is the only quality *improvement* parameter in the model (since the si_gain parameter is positive, a negative weight results in negative contributions to VQM which produce quality improvements). The si_gain parameter measures improvements to quality that result from edge sharpening or enhancement. Clipping of the parameter at an *upper* threshold of 0.14 immediately before multiplying by the parameter weight prevents excessive improvements to VQM of more than about 1/3 of a quality unit, which is the maximum improvement observed in the general subjective data set (i.e., an HRC will only be rewarded for a little edge enhancement).

The total VQM (after the contributions of all the parameters are added up) is clipped at a lower threshold of 0.0 to prevent negative VQM numbers. Finally, a crushing function that allows a maximum of 50% overshoot is applied to VQM values over 1.0 to limit VQM values for excessively distorted video that falls outside the range of the currently available subjective data.

If $VQM_G > 1.0$, then $VQM_G = (1 + c)*VQM_G / (c + VQM_G)$, where $c = 0.5$.

VQM_G computed in the above manner will have values greater than or equal to zero and a nominal maximum value of one. VQM_G may occasionally exceed one for video scenes that are extremely distorted.

Annex A
(Informative)

A Bibliography

[1] T1.801.01-1995 (R2001), *Digital Transport of Video Teleconferencing/Video Telephony Signals – Video Test Scenes for Subjective and Objective Performance Assessment*.[15]

[2] T1.801.02-1995 (R2001), *Digital Transport of Video Teleconferencing/Video Telephony Signals – Performance Terms, Definitions, and Examples*.[16]

[3] T1.801.03-1996, *Digital Transport of One-Way Video Signals – Parameters for Objective Performance Assessment*.[16]

[4] T1.801.04-1997 (R2002), *Multimedia Communications Delay, Synchronization, and Frame Rate Measurement*.[16]

[5] T1.TR.72-2001, *Methodological Framework for Specifying Accuracy and Cross-Calibration of Video Quality Metrics*.[16]

[6] T1.TR.73-2001, *Video Normalization Methods Applicable to Objective Video Quality Metrics Utilizing a Full Reference Technique*.[16]

[7] T1.TR.74-2001, *Objective Video Quality Measurement Using a Peak-Signal-to-Noise-Ratio (PSNR) Full Reference Technique*.[16]

[8] T1.TR.75-2001, *Objective Perceptual Video Quality Measurement Using a JND-Based Full Reference Technique*.[16]

[9] T1.TR.77-2002, *Data and sample program code to be used with the method specified in T1.TR.72-2001 for the calculation of resolving power of the video quality metrics in T1.TR.74-2001 and T1.TR.75-2001*.[16]

[10] ITU-R Recommendation BT.500, *Methodology for subjective assessment of the quality of television pictures,* Recommendations of the ITU, Radiocommunication Sector.[2]

[11] ITU-T Recommendation H.261, *"Video codec for audiovisual services at p x 64 kbit/sec,"* Recommendations of the ITU, Telecommunication Standardization Sector.[3]

[12] ITU-T Recommendation J.143, *"User requirements for objective perceptual video quality measurements in digital cable television,"* Telecommunication Standardization Sector.[3]

[13] ITU-T Recommendation P.910, *"Subjective video quality assessment methods for multimedia applications,"* Recommendations of the ITU, Telecommunication Standardization Sector.[3]

[14] ITU-T Recommendation P.931, *"Multimedia communications delay, synchronization, and frame rate measurement,"* Recommendations of the ITU, Telecommunication Standardization Sector.[3]

[15] ITU-T COM 9-80-E, *"Final report from the video quality experts group (VQEG) on the validation of objective models of video quality assessment,"* approved for release at VQEG meeting number 4, Ottawa, Canada, Mar. 2000.[3]

[16] A. K. Jain, *Fundamentals of Digital Image Processing*, Englewood Cliffs, NJ: Prentice-Hall Inc., 1989, pp. 348-357.

[17] SMPTE 125M, *"Television – Component Video Signal 4:2:2 – Bit-Parallel Digital Interface,"* Society of Motion Picture and Television Engineers, 595 West Hartsdale Avenue, White Plains, NY 10607.

[15] This document is available from the Alliance for Telecommunications Industry Solutions, 1200 G Street N.W., Suite 500, Washington, DC 20005. <http://www.atis.org>

[18] SMPTE 170M, "*SMPTE Standard for Television – Composite Analog Video Signal – NTSC for Studio Applications,"* Society of Motion Picture and Television Engineers, 595 West Hartsdale Avenue, White Plains, NY 10607.

[19] SMPTE Recommended Practice 187 – 1995, *"Center, Aspect Ratio, and Blanking of Video Images,"* Society of Motion Picture and Television Engineers, 595 West Hartsdale Avenue, White Plains, NY 10607.

[20] S. Wolf and M. Pinson, *"In-service performance metrics for MPEG-2 video systems,"* in Proc. Made to Measure 98 - Measurement Techniques of the Digital Age Technical Seminar, technical conference jointly sponsored by the International Academy of Broadcasting (IAB), the ITU, and the Technical University of Braunschweig (TUB), Montreux, Switzerland, Nov. 12-13, 1998.

[21] S. Wolf and M. Pinson, *"Spatial-temporal distortion metrics for in-service quality monitoring of any digital video system,"* in Proc. SPIE International Symposium on Voice, Video, and Data Communications, Boston, MA, Sep. 1999.

[22] S. Wolf and M. Pinson, *"The relationship between performance and spatial-temporal region size for reduced-reference, in-service video quality monitoring systems,"* in Proc. SCI / ISAS 2001 (Systematics, Cybernetics, and Informatics / Information Systems Analysis and Synthesis), Jul. 2001, pp. 323-328.

[23] S. Wolf and M. Pinson, *"Video Quality Measurement Techniques,"* NTIA Report 02-392, Jun. 2002.

[24] M. Pinson and S. Wolf, "*Video Quality Measurement User's Manual,"* NTIA Handbook 02-1, Feb. 2002.

[25] *Draft Final Report from the Video Quality Experts Group on the Validation of Objective Models of Video Quality Assessment, Phase II,* Video Quality Experts Group, Mar. 2003.