A New Standardized Method for Objectively Measuring Video Quality

Margaret H Pinson and Stephen Wolf

***Abstract*— The National Telecommunications and Information Administration (NTIA) General Model for estimating video quality and its associated calibration techniques were independently evaluated by the Video Quality Experts Group (VQEG) in their Phase II Full Reference Television (FR-TV) test. The NTIA General Model was the only video quality estimator that was in the top performing group for *both* the 525-line and 625-line video tests. As a result, the American National Standards Institute (ANSI) adopted the NTIA General Model and its associated calibration techniques as a North American Standard in 2003. The International Telecommunication Union (ITU) has also included the NTIA General Model as a normative method in two Draft Recommendations. This paper presents a description of the NTIA General Model and its associated calibration techniques. The independent test results from the VQEG FR-TV Phase II tests are summarized, as well as results from eleven other subjective data sets that were used to develop the method.**

***Index Terms*— Video Quality, Image Quality, objective testing, subjective testing.**

I. Introduction

The advent of digital video compression, storage, and transmission systems exposed fundamental limitations of techniques and methodologies that have traditionally been used to measure video performance. Traditional performance parameters relied on the "constancy" of a video system's performance for different input scenes. Thus, one could inject a test pattern or test signal (e.g., a static multi-burst), measure some resulting system attribute (e.g., frequency response), and be relatively confident that the system would respond similarly for other video material (e.g., video with motion). However, modern digital video systems adapt and change their behavior depending upon the input scene and the operational characteristics of the digital transmission system (e.g., bit-rate, error rate). Therefore, attempts to use input scenes that differ from what is actually used in-service (i.e., the actual user's video) can result in erroneous and misleading results.

Manuscript received November 19, 2003. This work was supported by the U.S. Department of Commerce.

M. H. Pinson is with the Institute for Telecommunication Sciences, Boulder, CO 80305 USA (phone: 303-497-3579; fax: 303-497-3680; e-mail: margaret@its.bldrdoc.gov).

S. Wolf is with the Institute for Telecommunication Sciences, Boulder, CO 80305 USA (e-mail: wolf@its.bldrdoc.gov).

NTIA pioneered perception-based video quality measurement in 1989 [1]. Subsequently, other organizations have performed major research efforts [2]-[10]. NTIA's research has focused on developing technology independent parameters that model how people perceive video quality. These parameters have been combined using linear models to produce estimates of video quality that closely approximate subjective test results. With the assistance of other organizations (e.g., VQEG), NTIA has collected data from 18 independent video quality experiments. The resulting 2944 subjectively rated video sequences were all sampled according to ITU-R Recommendation BT.601[1] [11]. This wide variety of input scenes and transmission systems has enabled NTIA to develop robust, technology independent parameters and video quality models.

This paper provides a description of the National Telecommunications and Information Administration (NTIA) General Model for estimating video quality and its associated calibration techniques (e.g., estimation and correction of spatial alignment, temporal alignment, and gain/offset errors). The General Model was metric H in the Video Quality Experts Group (VQEG) Phase II Full Reference Television (FR-TV) tests [12]. These algorithms have been standardized by the American National Standards Institute (ANSI) in the updated version of T1.801.03 [13], and have been included as a normative method in two International Telecommunication Union (ITU) recommendations [14][15].

The General Model was designed to be a general purpose video quality model (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 the VQEG Phase II test. While the independent VQEG Phase II FR-TV tests only evaluated the performance of the General Model on MPEG-2 and H.263 video systems, the General Model was developed using a wide variety of video systems and thus should work well for many other types of coding and transmission systems (e.g., bit rates from 10 kbits/s to 45 Mbits/s, MPEG-1/2/4, digital transmission systems with errors, analog transmission systems, and tape-based systems, utilizing both interlace and progressive video).

[1] 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. If 8 bits are used to uniformly sample the Y signal, Rec. 601 specifies that reference black be sampled at 16 and reference white at 235.

The General Model utilizes reduced-reference technology [16] and provides estimates of the overall impressions of video quality (i.e., mean opinion scores, as produced by panels of viewers). Reduced-reference measurement systems utilize low-bandwidth features that are extracted from the source and destination video streams. Thus, reduced-reference systems can be used to perform real-time in-service quality measurements (provided an ancillary data channel is available to transmit the extracted features), a necessary attribute for tracking dynamic changes in video quality that result from time varying changes in scene complexity and/or transmission systems. The General Model utilizes reduced-reference parameters that are extracted from optimally-sized spatial-temporal (S-T) regions of the video sequence. The General Model requires an ancillary data channel bandwidth of 9.3% of the uncompressed video sequence, and the associated calibration techniques require an additional 4.7%.

The General Model and its associated calibration techniques comprise a complete automated objective video quality measurement system (see Fig. **1**). The calibration of the original and processed video streams includes spatial alignment, valid region estimation, gain & level offset calculation, and temporal alignment. VQM calculation involves extracting perception-based features, computing video quality parameters, and combining parameters to construct the General Model. This paper will first provide a summary description of each process (the reader is referred to [17] for a more detailed description). Finally, test results from eleven subjective data sets and the independent VQEG FR-TV Phase II tests will be presented.

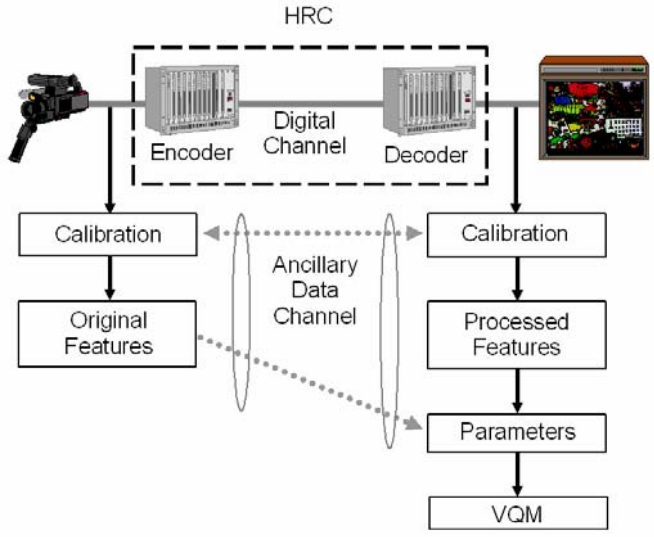

Fig. 1. Block Diagram of entire VQM

II. Spatial Alignment

The spatial alignment process determines the horizontal and vertical spatial shift of the processed video relative to the original video. The accuracy of the spatial alignment algorithm is to the nearest 0.5 pixel for horizontal shifts and to the nearest line for vertical shifts. After the spatial alignment 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 interlaced video, this may include reframing of the processed video stream as implied by comparison of the vertical field one and field two shifts. 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). Reframing impacts spatial alignment for both 525-line video (e.g., NTSC) and 625-line video (e.g., PAL) identically and causes the field-two vertical shift (in field lines) to be one greater than the field-one vertical shift (in field lines).

Spatial alignment must be determined before the processed valid region (abbreviated as PVR and defined as that portion of the processed video image which contains valid picture information), gain and level offset, and temporal alignment. 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 other calibration estimates would be corrupted. Unfortunately, spatial alignment cannot be correctly determined unless the PVR, gain and level offset, and temporal alignment are also known. The interdependence of these quantities produces a "chicken or egg" measurement problem. Calculation of the spatial alignment 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. Specifying the above quantities too narrowly could result in spatial alignment errors. The solution presented here performs an iterative search to find the closest matching original frame for each processed frame.[2]

An initial baseline (i.e., starting) estimate for vertical shift, horizontal shift, and temporal alignment is computed for one processed frame using a multi-step search. 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 frame. Gain compensation is not considered in the broad search, and PVR is set to exclude the over-scan portion of the picture which, in most cases, will eliminate invalid video from being used. The second step is a broad search for the approximate spatial shift, performed using a more limited range of original frames. The broad search for spatial shift covers approximately two dozen spatial shifts. Fewer downward

[2] When operating on interlaced 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 calibration algorithms will be entirely described for progressive video, this being the simpler case.

shifts are considered, since these are less likely to be encountered in practice. The third step performs localized spatial-temporal searches to fine-tune the spatial and temporal estimates. Each fine search includes a small set of spatial shifts centered around the current spatial alignment estimate and just three frames temporally, centered around the current best matching original frame. The zero shift condition is included as a safety check that helps prevent the algorithm from wandering and converging to a local minimum. This third step iterates up to five times. If these repeated fine searches fail to find a stable result (i.e., a local minimum), the above procedure is repeated using a different processed frame. This produces a baseline estimate that will be updated periodically, as described below.

The spatial alignment algorithm calculates the spatial alignment for each of a series of processed frames at some specified frequency (e.g., one frame every half-second). Using the baseline estimate as a starting point, the algorithm performs alternate fine searches (as described above) and estimations of the luminance gain and level offset. To calculate the luminance gain and level offset, the mean and standard deviation of the original and processed frames are compared, using the current spatial and temporal alignment estimates. This simple calculation has a robust performance in the presence of alignment errors. If the baseline estimate is correct or very nearly correct, three fine searches will normally yield a stable result. 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). In this case, 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 shift 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, alternating with luminance gain and level offset calculations. If this second repeated fine search fails to find a stable result, then spatial alignment has failed for this frame. If a stable result has been found for this frame, the spatial shifts (i.e., horizontal and vertical) are stored and the baseline estimate is updated.

For some processed frames, the spatial alignment algorithm could fail. Usually, when the spatial alignment is incorrectly estimated for a processed frame, the ambiguity is due to characteristics of the scene. Consider, for example, a digitally created progressive scene containing a pan to the left. Because the pan was computer generated, this scene could have a horizontal pan of exactly two pixels every frame. From the spatial alignment search algorithm's point of view, it would be impossible to differentiate between the correct spatial alignment computed using the matching original frame, and a two pixel horizontal shift computed using the frame that occurs one frame prior to the matching original frame. 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 acceptable.

Therefore, the iterative search algorithm should be applied to a sequence of processed frames. The individual estimates of horizontal and vertical shifts from multiple processed frames are then median-filtered to produce more robust estimates. Using the 50^{th} percentile point allows the most likely horizontal and vertical shift to be chosen. This algorithm consistently produces a horizontal spatial alignment accuracy that is good to the nearest 0.5 pixels.[3] Spatial shift estimates from multiple sequences or scenes may be further combined to produce an even more robust estimate for the Hypothetical Reference Circuit (HRC)[4] being tested, assuming that the spatial shift is constant for all scenes passing through the HRC.

The spatial alignment algorithm described above requires a relatively high ancillary data channel bandwidth, due to the pixel-by-pixel comparison of original and processed frames. This could impact the design of an in-service quality monitoring application. Fortunately, each piece of video transmission equipment (i.e., encoder, decoder, or analog transmission) will normally have one constant spatial alignment. If the hardware had a changing or variable spatial alignment, the transmitted video would appear to move up and down – an unacceptable degradation that would be quickly addressed by the manufacturer.

III. Processed Valid Region (PVR)

Video sampled according to ITU-R Recommendation BT.601 [11], henceforth abbreviated as Rec. 601, may have a border of pixels and lines that does 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 occur in the displayed picture area, the viewer may notice a black border around the displayed images, since the video 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 contain several lines of corrupted video at the top or bottom that may not be visible to the viewer (e.g., VHS tape recorders can corrupt several lines at the bottom of the picture in the over-

[3] Spatial alignment to the nearest 0.5 pixels is sufficient for the video quality measurements described herein.

[4] The term HRC is used here to denote one instantiation of a video transmission system which may include an encoder, a digital transmission system, and a decoder. HRC is a generic term commonly used by standards bodies to protect the anonymity of video equipment suppliers.

scan area).

To prevent non-picture areas from influencing the VQM measurements, these areas are excluded from the VQM measurement. Since the behavior of some video systems is scene dependent, the valid region should ideally be calculated using actual video streams. In this case, PVR should be calculated for each scene separately. After PVR has been calculated, the invalid pixels are discarded from the original and processed video sequences.

The automated valid region algorithm estimates the valid region of the processed video stream 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. The core algorithm starts with an assumption that the outside edges of each processed frame contain invalid video. The extent of this invalid region is set empirically, based upon observations of actual video systems. For 525-line video sampled according to Rec. 601, the default invalid region excludes 6 pixels/lines at the top, left and right, and 4 lines at the bottom. The PVR algorithm begins by setting the PVR to exclude this default invalid region. The pixels immediately inside the current valid region estimate are then examined. If the average pixel value is black or ramping up slowly from black, then the valid region estimate is accordingly decreased in size. By repeating this examination, the valid region is iteratively diminished in size.

The stopping conditions 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 the processed video sequence and the largest observed PVR (with some safety margin added) is used for the final PVR estimate. The coordinates of the PVR are transformed via the spatial alignment results, so that the PVR specifies the portion of the original video that remains valid.

This automated valid region algorithm works well to estimate the valid region of most scenes. Due to the nearly infinite possibilities for scene content, the algorithm 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 portion of the video. On the other hand, including corrupted video in the video quality calculations may have a large impact on the quality estimate.

The valid region algorithm can also be applied to the original video sequence. The resulting original valid region (OVR) increases the accuracy of the processed valid region calculation, by providing a maximal bound on the PVR.

IV. Gain & Level Offset

A prerequisite for performing gain and level offset calibration is that the original and processed images be spatially registered. The original and processed images must also be temporally registered, which will be addressed later. 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). In addition, since a linear estimation algorithm is utilized, excessive gains that cause pixel levels to be clipped will cause estimation errors unless the algorithm is modified to allow for this effect.

The valid regions of the original and processed frames are divided into small, square sub-regions, or blocks. The mean over space of the [Y, C_B, C_R] samples for each corresponding original and processed sub-region are computed to form spatially sub-sampled images. To temporally register a processed frame (with spatial shift held constant), the standard deviation of each (original minus processed) difference image is computed using the sub-sampled Y luminance frames. For a given processed frame, the temporal shift that produces the smallest standard deviation (i.e., most cancellation with the original) is chosen as the best match. A first order linear fit is used to compute the relative gain and offset between the sub-sampled original and processed frames. This linear fit is applied independently to each of the three channels: Y, C_B, and C_R.

The algorithm described above should be applied to multiple matching original and processed frame pairs distributed at regular intervals throughout the video sequence. 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 the level offset and gain is constant for all scenes that have passed through a given HRC, then measurements performed on each scene can be filtered (across all the scenes) to increase robustness and accuracy. The overall HRC level offset and gain results can then be used to compensate all of the processed video for that HRC.

Although gain and level offsets are calculated for the C_B and C_R channels, these correction factors are *not* applied. The General Model utilizes only the luminance or Y channel gain and level offset correction factors. Changes to the C_B and C_R color channels are considered impairments for which the system under test should be penalized.

V. Temporal Alignment

Modern digital video communication systems typically require several tenths of a second to process and transmit the video from the sending camera to 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 and service providers for specification and comparison of interactive services. 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 ideally be made in-service to be truly representative and accurate. Estimates of video delay are required to temporally align the original and processed video streams 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 timing information). In general, however, time synchronization between the original and processed video streams must be measured. This section 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.

To reduce the influence of distortions on temporal alignment, original and processed images are spatially sub-sampled and then normalized to have unit variance. Each individual processed image is then temporally registered using the technique presented for the gain & level offset algorithm (i.e., find the original image that minimizes the standard deviation of the difference between the original and processed images). This locates the most similar original image for each processed image.

However, it is not the identity of the original image that is of interest, but rather the *relative delay* between the original and processed images (e.g., in seconds or frames). The delay measurements from a series of images are combined into a histogram, which is then smoothed. If a bin near one end of the histogram contains a large number count, then the temporal alignment uncertainty was too small and the entire temporal alignment algorithm should be re-run with a larger temporal uncertainty. Otherwise, the maximum smoothed histogram bin indicates the best average temporal alignment for the scene. This counting scheme produces an accurate estimate for the average delay of a video sequence.

Unlike the previous calibration algorithms, the temporal alignment algorithm examines every processed video frame. Some of these individual temporal alignment measurements may be incorrect but those errors will tend to be randomly distributed. Patterns in the histogram can provide insights into the system under test, such as an indication of changing or variable delay. Delay measurements from still and nearly motionless portions of the scene are not used, since the original images are nearly identical to each other.

The delay indicated at the final stage of the algorithm may be different from the delay a viewer might choose when 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 are then examined to determine the extent and statistics of any variable video delay present in the HRC.

VI. An Overview of Feature and Parameter Calculation Methods

A quality *feature* in the context of this algorithm 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 are a function of space and time that characterize perceptual changes in the spatial, temporal, and chrominance properties of video streams. By comparing features extracted from the calibrated processed video with features extracted from the original video, quality *parameters* can be computed that are indicative of perceptual changes in video quality.

Viewed conceptually, all of the features used by the General Model perform the same steps. 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.

All perceptual filters operate on frames within a calibrated video sequence. Thus, the pixels in original and processed images outside of the PVR have been discarded, the processed sequence has been spatially registered, the processed luminance Y images have been gain/level offset compensated, and the processed sequence has been temporally registered. All features operate independently of image size (i.e., S-T region size does not change when the image size changes).[5]

Each perceptual filter distinguishes some aspect of video quality. The luminance image plane contains information pertinent to edge business and noise. An edge enhanced version of the luminance Y image plane more accurately identifies blurring, blocking, and other large-scale edge effects. The color image planes, C_B and C_R, are useful for identifying hue impairments and digital transmission errors. Time differencing consecutive luminance Y image planes highlights jerky or unnatural motion.

After the original and processed video streams have been

[5] This independence of S-T region size and image size has only been tested for standard definition television, including CIF and QCIF sequences. We

perceptually filtered, the video streams are divided into abutting S-T regions. 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. Since the processed video has been calibrated, for each processed S-T region there exists a corresponding original S-T region. Features are extracted from each of these S-T regions using a simple mathematical function. The two functions that work best are mean, which measures the average pixel value, and standard deviation, which estimates the spread of pixel values. After feature extraction, the temporal axis no longer relates to individual frames. The temporal extent of the S-T regions determines the sample rate of the feature stream. This sample rate cannot exceed the frame rate.

Finally, some feature values are clipped to prevent them from measuring impairments that are imperceptible. This clipping is of the form:

$$f_{clip} = \max(f, threshold) \tag{1}$$

where f is the feature before clipping, *threshold* is the clipping threshold, and f_{clip} is the feature after clipping. Since clipping is applied to both the original and processed feature streams, this clipping serves to reduce sensitivity to imperceptible impairments.

Where quality *features* quantify some perceptual aspect of one video stream, quality *parameters* compare original and processed features to obtain an overall measure of video distortion. Viewed conceptually, all of the parameters used by the General Model perform the same steps. First, the processed feature value for each S-T region is compared to the corresponding original feature value using comparison functions that emulate the perception of impairments. Next, perception-based error-pooling functions are applied across space and time. Error pooling across space will be referred to as spatial collapsing, and error pooling across time will be referred to as temporal collapsing. Sequential application of the spatial and temporal collapsing functions to the streams of S-T quality parameters produces single-value quality parameters for the entire video sequence, which is nominally 8 to 10 seconds in duration. [6] The final space-time collapsed parameter values may also be scaled and clipped to account for nonlinearities and to better match the parameter's sensitivity to impairments with the human perception of those impairments.

The perceptual impairment at each S-T region is calculated using comparison functions that have been developed to model visual masking of spatial and temporal impairments. Some features use a comparison function that performs a simple Euclidean distance between two original and two processed feature streams.

$$p = \sqrt{(f_o - f_p)^2 + (f_{o2} - f_{p2})^2} \tag{2}$$

However, most features use either the ratio comparison function

$$p = (f_p - f_o) / f_o \tag{3}$$

or the log comparison function

$$p = \log 10\left(\frac{f_p}{f_o}\right) \tag{4}$$

where f_o and f_{o2} are original feature values; f_p and f_{p2} are the corresponding processed feature values.

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).

The ratio and log comparison functions produce a mixture of positive and negative values, where positive numbers indicate gains, and negative numbers indicate losses. Greater measurement accuracy can be obtained by examining losses and gains separately. The fundamental reason is that humans generally react more negatively to additive impairments (e.g., blocking which produces extra edges) than subtractive impairments (e.g., blurring which produces a loss of edge sharpness) and hence losses and gains must be given different weights in the quality estimator. Therefore, the ratio and log comparison functions are always followed by either a loss function (i.e., replace positive values with zero) or a gain function (i.e., replace negative values with zero).

The parameters from the S-T regions form three-dimensional arrays spanning the temporal axis and two spatial dimensions (i.e., horizontal and vertical). For the spatial collapsing step, impairments from the S-T regions with the same time index are pooled using a spatial collapsing function (e.g., mean, standard deviation, or rank-sorting with percent threshold selection). Spatial collapsing yields a time history of parameter values. Extensive investigations performed by NTIA revealed that the optimal spatial collapsing function often involves some form of worst case processing, such as taking the average of the worst 5% of the distortions observed at that point in time. 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.

The parameter time history results from the spatial collapsing function are next pooled using a temporal collapsing function to produce an objective parameter for the video sequence. Viewers use a variety of temporal collapsing functions. For example, the *mean* over time is indicative of the average quality that is observed during the time period. The *90%* level for a gain parameter's time history is indicative of the worst additive transient impairment that is observed (e.g., digital transmission errors may cause a 1 to 2 second disturbance in the processed video).

The all-positive or all-negative temporally collapsed

---

expect high definition television (HDTV) to exhibit this independence as well, but this has not been tested.

[6] Most of the video sequences that were used to develop the General Model were from 8 to 10 seconds in duration.

parameters 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, since the linear least-squares algorithm will be used to determine the optimal parameter weights. Two nonlinear scaling functions that might be applied are the square root function, and the square function. If the square root function is applied to an all-negative parameter, the parameter is first made all positive (i.e., absolute value taken).

Finally, a clipping function might be applied to reduce the parameter's sensitivity to small amounts of impairment. This clipping function for positive parameters is:

$$p' = \begin{cases} 0 & \text{if } p \le t \\ p - t & \text{otherwise} \end{cases} \quad (5)$$

where t is the threshold.

When designing individual parameters, the specific details of each step are established by analyzing subjectively rated video. For example, *threshold* from equation (1) and p from equation (5) are set to values that maximize the correlation between the quality parameter and subjective video quality ratings. Thus, the specific details of the General Model parameters were chosen to best emulate human perception.

VII. General Model Parameters

The General Model contains seven independent parameters. Four parameters are based on features extracted from spatial gradients of the Y luminance component, two parameters are based on features extracted from the vector formed by the two (C_B, C_R) chrominance components, and one parameter is based on the product of features that measure contrast and motion, both of which are extracted from the Y luminance component. The seven parameters are computed as described below.

A. Parameter "si_loss"

Parameter **si_loss** detects a decrease or loss of spatial information (e.g., blurring). This parameter uses a 13 pixel spatial information filter (SI13) that has a peak response at approximately 4.5 cycles/degree (when Rec. 601 video is viewed at a distance of 6 times picture height). The SI13 filter was specifically developed to measure perceptually significant edge impairments [17]. An alternate method for extracting edges is the Sobel filter, but the 3 pixel Sobel filter detects details so fine that people may not care if they are blurred. SI13 utilizes 13 pixel by 13 pixel horizontal and vertical filter masks. These two filter masks are created by horizontal and vertical replication of the following vector:

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

The horizontal and vertical filters are separately applied to the luminance image. The resulting filtered images (I_H and I_V) are combined into a single image (I_{SI13}) using Euclidean distance (i.e., square root of the sum of the squares).

The **si_loss** parameter is calculated by performing the following seven steps:

1) Apply the SI13 filter to each luminance image.
2) Divide each video sequence into 8 pixel x 8 line x 0.2 second S-T regions. This is the optimal S-T region size for the **si_loss** parameter [18].
3) Compute the standard deviation of each S-T region.
4) Apply a perceptibility threshold, replacing values less than 12 with 12.
5) Compare original and processed feature streams (each computed using steps 1 through 4) using the ratio comparison function (see equation 3) followed by the loss function.
6) Spatially collapse by computing the average of the worst (i.e., most impaired) 5% of S-T blocks for each 0.2 second slice of time.
7) Temporally collapse by sorting values in time and selecting the 10% level. Since the parameter values are all negative, this is a form of worst-case temporal processing.

B. Parameter "hv_loss"

The **hv_loss** parameter detects a shift of edges from horizontal & vertical orientation to diagonal orientation, such as might be the case if horizontal and vertical edges suffer more blurring than diagonal edges. This parameter uses the horizontally and vertically filtered images (H and V) output from the SI13 filter. Two new perceptually filtered images are created: one contains horizontal and vertical edges (HV) and the other contains diagonal edges (HV_{BAR}, or complement of HV). An edge angle is computed for each pixel by taking the four-quadrant arctangent of the SI13 filtered H and V pixel values. The HV image contains values where the angle is within 0.225 radians of horizontal or vertical, and zero otherwise. The HV_{BAR} image contains values where the angle indicates a diagonal edge, and zero otherwise. Pixels with an SI13 magnitude value less than 20 are not used (i.e., replaced with zero), because the angle calculation is unreliable.

The **hv_loss** parameter is calculated by performing the following nine steps:

1) Apply the HV and HV_{BAR} perceptual filters to each luminance plane.
2) Divide each of the HV and HV_{BAR} video sequences into 8 pixel x 8 line x 0.2 second S-T regions. This is the optimal S-T region size for the **hv_loss** parameter [18].
3) Compute the mean of each S-T region.
4) Apply a perceptibility threshold, replacing values less than 3 with 3.
5) Compute the ratio (HV / HV_{BAR}).
6) Compare original and processed feature streams (each computed using steps 1 through 5) using the ratio comparison function (see equation 3) followed by the loss function.
7) Spatially collapse by computing the average of the worst 5% of blocks for each 0.2 second slice of time.
8) Temporally collapse by taking the mean over all time slices.
9) Square the parameter (i.e., non-linear scaling), and clip at a minimum value of 0.06 (see equation 5).

Due to the non-linear scaling, the values associated with parameter **hv_loss** are all positive, rather than being all negative as is the case for the other loss metric, **si_loss**.

C. Parameter "hv_gain"

This parameter detects a shift of edges from diagonal to horizontal & vertical, such as might be the case if the processed video contains tiling or blocking artifacts.

1) Perform steps 1 through 5 from parameter **hv_loss**.
2) Compare original and processed feature streams using the log comparison function (see equation 4) followed by the gain function.
3) Spatially collapse by computing the average of the worst 5% of blocks for each 0.2 second slice of time.
4) Temporally collapse by taking the mean over all time slices.

D. Parameter "chroma_spread"

This parameter detects changes in the spread of the distribution of two-dimensional color samples.

1) Divide the C_B and C_R color planes into separate 8 pixel x 8 line x 1 frame S-T regions.
2) Compute the mean of each S-T region. Multiple the C_R means by 1.5 to increase the perceptual weighting of the red color component in the next step.
3) Compare original and processed feature streams C_B and C_R using Euclidean distance (see equation 2).
4) Spatially collapse by computing the standard deviation of blocks for each 1-frame slice of time.
5) Temporally collapse by sorting the values in time and selecting the 10% level, and then clip at a minimum value of 0.6. Since all values are positive, this represents a best-case processing temporally. Thus, **chroma_spread** measures color impairments that are nearly always present.

Steps 1 and 2 essentially sub-sample the C_B and C_R image planes. Just as **si_loss**, **hv_loss**, and **hv_gain** examine edges containing enough pixels to be perceptually significant, **chroma_spread** performs coherent integration (i.e., C_B and C_R treated as a vector) of color samples over an area large enough to have significant perceptual impact.

E. Parameter "si_gain"

This is the only quality *improvement* parameter in the model. The **si_gain** parameter measures improvements to quality that result from edge sharpening or enhancements. The **si_gain** parameter is calculated by performing the following five steps:

1) Perform steps 1 through 3 from **si_loss**.
2) Apply a perceptibility threshold, replacing values less than 8 with 8.
3) Compare original and processed feature streams (each computed using steps 1 and 2) using the log comparison function paired with the gain function.
4) Spatially and temporally collapse by computing the average of all blocks, and then clip at a minimum value of 0.004. These steps estimate the average overall level of edge enhancement that is present.
5) Set all values greater than 0.14 equal to 0.14 to prevent excessive quality improvements of more than about one-third of a quality unit when multiplied by the parameter weight (see section VIII). One-third of a quality unit is the maximum improvement observed in the subjective data that was used to develop this parameter. Thus, an HRC will only be rewarded for a small amount of edge enhancement. The **si_gain** parameter is a relative enhancement in quality for systems that perform contrast enhancement with respect to systems that don't perform contrast enhancement. In Section VIII, VQM will be clipped to prevent the **si_gain** parameter from producing processed quality estimates better than the original.

F. Parameter "ct_ati_gain"

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 product of contrast information and temporal information can be used to partially account for these interactions. The **ct_ati_gain** metric is computed as the product of a contrast feature, measuring the amount of spatial detail, and a temporal information feature, measuring the amount of motion present in the S-T region. Impairments will be more visible in S-T regions that have a low product than in S-T regions that have a high product. This is particularly true of impairments like noise and error blocks. **ct_ati_gain** identifies moving-edge impairments that are nearly always present, such as edge noise.

1) Apply the "absolute value of temporal information" (ATI) motion detection filter to each luminance plane. ATI is the absolute value of a pixel-by-pixel difference between the current and previous video frame.
2) Divide each video sequence into 4 pixel x 4 line x 0.2 second S-T regions.
3) Compute the standard deviation of each S-T region.
4) Apply a perceptibility threshold, replacing values less than 3 with 3.
5) Repeat steps 2 through 4 on the Y luminance video sequence (without perceptual filtering) to form "contrast" feature streams.
6) Multiply the contrast and ATI feature streams.
7) Compare original and processed feature streams (each computed using steps 1 through 6) using the ratio comparison function (see equation 3) followed by the gain function.
8) Spatially collapse by computing the mean of each 0.2 second slice of time.
9) Temporally collapse by sorting values in time and selecting the 10% level. The parameter values are all positive, so this temporal collapsing function is a form of best-case processing, detecting impairments that are nearly always present.

G. Parameter "chroma_extreme"

This feature uses the same color features as the **chroma_spread** metric, but different spatial-temporal

collapsing functions. **Chroma_extreme** detects severe localized color impairments, such as those produced by digital transmission errors.

1) Perform steps 1 through 3 from **chroma_spread**.
2) Spatially collapse by computing for each slice of time the average of the worst 1% of blocks (i.e., rank-sorted values from the 99% level to the 100% level), and subtract from that result the 99% level. This identifies very bad distortions that impact a small portion of the image.
3) Temporally collapse by computing standard deviation of the results from step 2.

VIII. General Model

This section describes how to compute the General Model using the calculated parameter values. 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 the video quality parameters described in section VII. The General Model produces output values that range from zero (no perceived impairment) to approximately one (maximum perceived impairment). The General Model values may be multiplied by 100 to approximately scale results to the Difference Mean Opinion Score (DMOS) derived from the 100-point double stimulus continuous quality scale (DSCQS). The General Model was designed using Rec. 601 video that was subjectively evaluated at a viewing distance of four to six times picture height.

The General Video Quality Model (VQM) consists of the following linear combination of the seven parameters given in section VII:

VQM = - 0.2097 * **si_loss**
+ 0.5969 * **hv_loss**
+ 0.2483 * **hv_gain**
+ 0.0192 * **chroma_spread**
- 2.3416 * **si_gain**
+ 0.0431 * **ct_ati_gain**
+ 0.0076 * **chroma_extreme**

Note that **si_loss** is always less than or equal to zero, so **si_loss** can only *increase* VQM. Since all the other parameters are greater than or equal to zero, **si_gain** is the only parameter that can *decrease* VQM.

After the contributions of all the parameters are weighted and added up, VQM is clipped at a lower threshold of 0.0. This prevents **si_gain** values from producing a quality rating that is better than the original (i.e., a negative VQM). Finally, a crushing function that allows a maximum 50% overshoot is applied to VQM values over 1.0. The purpose of the crushing function is to limit VQM values for excessively distorted video that falls outside the range of the subjective data used to develop the model.

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

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

IX. Performance

The fundamental purpose of the General Model and the associated calibration routines is to track subjective video quality scores. This ability will be demonstrated by comparing General Model results with subjectively rated video clips.

A. Training Data

The General Model was developed using subjective and objective test data from eleven different video quality experiments. These eleven subjective experiments were conducted from 1992 to 1999. All of the data sets were collected in accordance with the most recent version of ITU-R Recommendation BT.500 [19] or ITU-T Recommendation P.910 [20] that was available when the experiment was performed. All of the data sets used scenes from 8 to 10 seconds in duration. Nine of the data sets (i.e., data sets one to nine) used double stimulus testing where viewers saw both the original and processed sequences. Two of the data sets (i.e., data sets ten and eleven) used single stimulus testing where viewers saw only the processed sequence. Seven of the data sets were primarily television experiments (i.e., data sets one to seven) while four of the data sets were primarily videoconferencing experiments (i.e., data sets eight to eleven). The subjective scores from each of the subjective data sets have been linearly mapped onto a common scale with a nominal range of [0,1] using the iterative nested least squares algorithm (INLSA) [17] [21] [22] and the seven parameters from the General Model. The reader is directed to [17] for more complete descriptions of these subjective experiments.

Taken together, these experiments include 1536 subjectively rated video sequences. Fig. 2 shows the scatter plot of subjective quality versus VQM, where each data set's video sequences are plotted in a different color (1 = black, 2 = red, 3 = green, 4 = blue, 5 = yellow, 6 = magenta, 7 = cyan, 8 = gray, 9 = dark red, 10 = copper, 11 = aquamarine). The Y-axis of Fig. 2 shows the subjective common scale. The overall Pearson linear correlation coefficient between subjective quality and VQM for the video sequences plotted in Fig. 2 is 0.948.

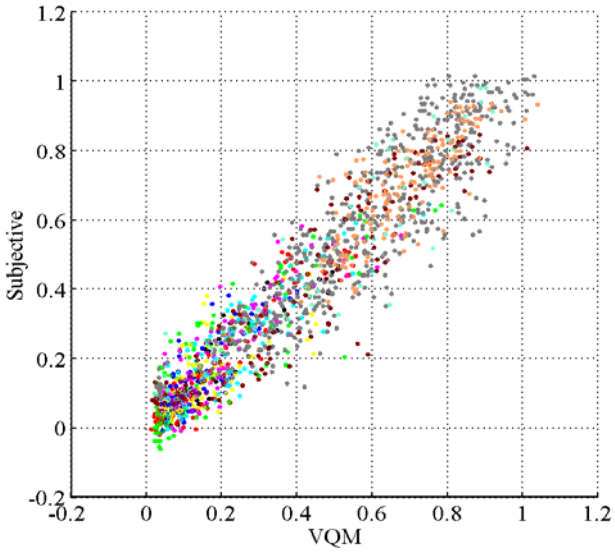

Fig. 2. Training data: clip subjective quality vs. clip VQM.

Fig. 3 shows the effect of averaging over scenes to produce a single subjective score (i.e., HRC subjective quality) and objective score (i.e., HRC VQM) for each video system. HRC subjective quality is indicative of how the system responds (on average) to a set of video scenes. The overall Pearson linear correlation coefficient between HRC subjective quality and HRC VQM for the data points in Fig. 3 is 0.980. For making video system (i.e., HRC) comparisons, the estimate of HRC subjective quality provided by HRC VQM is more accurate than the estimate of clip subjective quality provided by clip VQM. This can be seen by comparing the amount of scatter in Fig. 3 with the amount of scatter in Fig. 2.

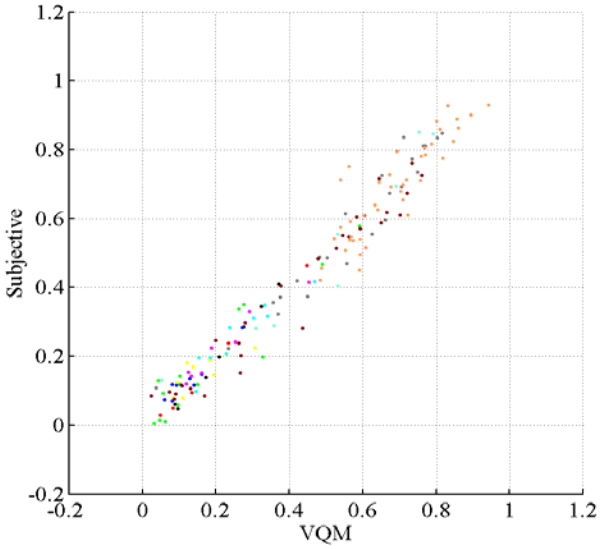

Fig. 3. Training data: HRC subjective quality vs. HRC VQM.

B. Testing Data

VQEG provides input to standardization bodies responsible for producing International Recommendations regarding objective video quality metrics. To that end, VQEG performed the FR-TV Phase II test, in 2001 to 2003 [12]. The VQEG FR-TV Phase II tests provided an independent evaluation of the ability of video quality models and their associated calibration algorithms to reproduce subjective scores. This test contained two experiments, one restricted to 525-line video and the other restricted to 625-line video. The subjective testing was performed by three independent labs.

The subjective data and VQM for these two experiments are plotted in Fig. 4 and Fig. 5. In the 525-line test, the General Model was one of only two models that performed statistically better than the other models tested. The Pearson linear correlation was 0.938, and the outlier ratio 0.46.[7] In the 625-line test, the General Model was one of four models that performed statistically better than the other models. The Pearson linear correlation was 0.886, and the outlier ratio 0.31. No model performed statistically better than the General Model in either the 525-line or 625-line test. All other models performed statistically worse than the General Model in either the 525-line or the 625-line test or both.

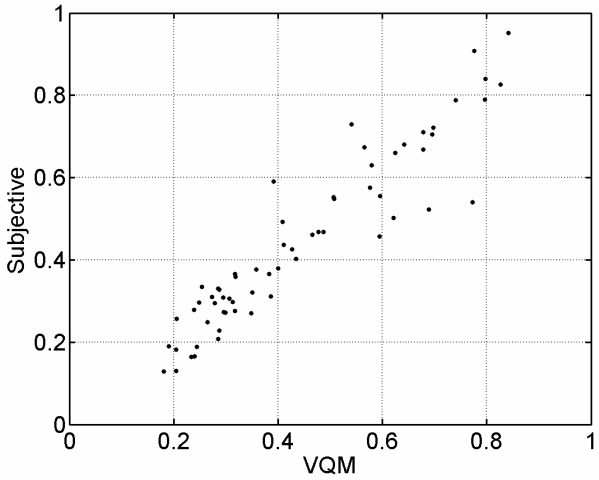

Fig. 4. 525-line VQEG FR-TV Phase II test data: clip subjective quality vs. clip VQM.

The data depicted in these two scatter plots are identical to that reported in [17]. The VQM scores from the General Model plotted in the VQEG graphs are not exactly equivalent to VQM values as given in section VIII. This is because the VQEG FR-TV Phase II data analysis applied a logistic transformation to each objective metric to remove non-linearities that might be present. However, the logistic transformation had only a minor impact on the VGM values, because these values exhibited a near-linear relationship to the VQEG FR-TV Phase II subjective test data. If the logistics transformation is not performed, the Pearson linear correlations are 0.930 for the 525-line test and 0.865 for the 625-line test.

[7] Outliers are data points with an error in excess of twice the standard error of the mean. The "outlier ratio" is the number of outliers divided by the total number of data points.

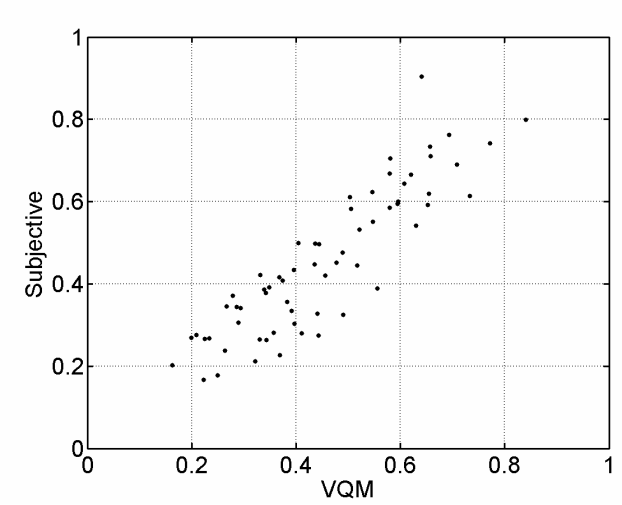

Fig. 5. 625-line VQEG FR-TV Phase II test data: clip subjective quality vs. clip VQM.

X. Conclusion

We have presented an overview of a general purpose video quality model (VQM) and its associated calibration routines. This model has been shown by the VQEG FR-TV Phase II test to produce excellent estimates of video quality for both 525-line and 625-line video systems. In the 525-line test, VQM was one of only two models that performed statistically better than the other models submitted for independent evaluation. In the 625-line test, VQM was one of four models that performed statistically better than the others. Overall, VQM was the only model that performed statistically better than the others in both the 525-line and 625-line tests. Obtaining an average Pearson correlation coefficient over both tests of 0.91, VQM was the only model to break the 0.9 threshold. As a result, VQM was standardized by ANSI in July 2003 (ANSI T1.801.03-2003), and has been included in Draft Recommendations from ITU-T Study Group 9 and ITU-R Working Party 6Q.

VQM and its associated automatic calibration algorithms have been completely implemented in user friendly software. This software is available to all interested parties via a no-cost license agreement [23].

References

[1] S. Wolf, "Features for automated quality assessment of digitally transmitted video," NTIA Report 264, June 1990. Available at www.its.bldrdoc.gov/n3/video/pdf/ntia264.pdf

[2] D. Hands, "A basic multimedia quality model," IEEE Transactions on Multimedia, to be published in 2004.

[3] A. Hekstra et al., "PVQM – A perceptual video quality measure," Signal Processing Image Communication 17, 2002, pp. 781-798.

[4] S. Winkler and R. Campos, "Video quality evaluation for internet streaming applications," in Proceedings of SPIE-IS&T Electronic Imaging, SPIE Vol. 507, 2003, pp. 104-115.

[5] C. Lee and O. Kwon, "Objective measurements of video quality using the wavelet transform," Optical Engineering v. 42 no 1, Jan. 2003, pp. 265-72.

[6] A. Worner, "Realtime quality monitoring of compressed video signals," SMPTE Journal v. 111 no 9, Sept 2002, pp. 373-7.

[7] H. Ikeda, T. Yoshida, and T. Kashima, "Mixed variables modeling method to estimate network video quality," SPIE Video Communications and Image Processing Conference, Lugano, Switzerland, Jul. 8-11 2003.

[8] J. Caviedes and F. Oberti, "No-reference quality metric for degraded and enhanced video," SPIE Video Communications and Image Processing Conference, Lugano, Switzerland, Jul. 8-11 2003.

[9] A. Pessoa, A. Falcão, A. Faria, and R. Lotufo, "Video quality assessment using objective parameters based on image segmentation," IEEE Int. Telecommunications Symposium 98, 1, Brazil, 1998, p. 498-503.

[10] A. Watson, J. Hu and J. McGowan, "DVQ: A digital video quality metric based on human vision," Journal of Electronic Imaging, 10(1), pp 20-29.

[11] ITU-R Recommendation BT.601, "Encoding parameters of digital television for studios," Recommendations of the ITU, Radiocommunication Sector.

[12] Video Quality Experts Group (VQEG), "Final report from the Video Quality Experts Group on the validation of objective models of video quality assessment, phase II," 2003 VQEG. Available: www.vqeg.org

[13] ANSI T1.801.03 – 2003, "American National Standard for Telecommunications – Digital transport of one-way video signals – Parameters for objective performance assessment," American National Standards Institute.

[14] Preliminary Draft New Recommendation "Objective perceptual video quality measurement techniques for digital broadcast television in the presence of a full reference," Recommendations of the ITU, Radiocommunication Sector.

[15] Draft Revised Recommendation J.144, "Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference," Recommendations of the ITU, Telecommunication Standardization Sector.

[16] ITU-T Recommendation J.143, "User requirements for objective perceptual video quality measurements in digital cable television," Recommendations of the ITU, Telecommunication Standardization Sector.

[17] S. Wolf and M. Pinson, "Video quality measurement techniques," NTIA Report 02-392, June 2002. Available: www.its.bldrdoc.gov/n3/video/documents.htm

[18] 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.

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

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

[21] M. Pinson and S. Wolf, "An objective method for combining multiple subjective data sets," SPIE Video Communications and Image Processing Conference, Lugano, Switzerland, Jul. 8-11 2003.

[22] S. D. Voran, "An iterated nested least-squares algorithm for fitting multiple data sets," NTIA Technical Memorandum TM-03-397, Oct. 2002. Available: www.its.bldrdoc.gov/home/programs/audio/pubs_talks.htm

[23] M. Pinson and S. Wolf, "Video quality metric software, Version 2," NTIA Software/Data Product SD-03-396, Volumes 1-5, Oct. 2002. Available: www.its.bldrdoc.gov/n3/video/vqmsoftware.htm

Margaret H. Pinson earned a B.S. and M.S. in Computer Science from the University of Colorado at Boulder, CO in 1988 and 1990, respectively. Since 1988 she has been working as a Computer Engineer at the Institute for Telecommunication Sciences (ITS), an office of the National Telecommunications and Information Administration (NTIA) in Boulder, Colorado. Her goal is to develop

automated metrics for assessing the performance of video systems and actively transfer this technology to end-users, standards bodies, and U.S industry. Her publications are available on-line at www.its.bldrdoc.gov/n3/video/documents.htm.

Stephen Wolf received a BS in electrical engineering from Montana State University at Bozeman, Montana in 1979 and an MS in electrical and computer engineering from the University of California at Santa Barbara, California in 1983. From 1979 until 1988, he worked on the design and development of radar signal processing and target recognition techniques, including highly advanced inverse synthetic aperture radar (ISAR) systems for the Naval Weapons Center in China Lake, CA. Since 1988, he has been Project Leader of the Video Quality Research Program at the Institute for Telecommunication Sciences (ITS), an office of the National Telecommunications and Information Administration (NTIA) in Boulder, Colorado. His contributions to the field include numerous papers and three U.S. patents for the development "reduced-reference" video quality measurement systems that emulate human perception. Mr. Wolf is an active participant and contributor to the standardization activities of IEEE, the American National Standards Institute (ANSI), and the International Telecommunication Union (ITU) and has served as Chief Technical Editor for video performance measurement standards and technical reports.