COMMITTEE T1
CONTRIBUTION

Document Number: T1A1.5/93-105

****

STANDARDS PROJECT: Analog Interface Performance Specifications for Digital Video Teleconferencing/Video Telephony Service

****

TITLE: Real-Time Measurement System for ITS Video Quality Parameters

****

ISSUE ADDRESSED: Real-time Processing

****

SOURCE: National Telecommunications and Information Administration
Institute for Telecommunication Sciences
(Coleen Jones)

****

DATE: 9 August 1993

****

DISTRIBUTION TO: T1A1.5

****

KEYWORDS: Video Quality, Video Performance Specifications, Objective Quality, Subjective Quality

****

DISCLAIMER:

****

1. Introduction

The Institute for Telecommunication Sciences has implemented two of its most effective video quality measures in a real-time personal computer-based measurement system. The two fundamental measures extracted by the real-time system characterize the spatial (*e.g.*, blurring) and temporal (*e.g.*, jerky motion) distortions in a video scene. The video quality parameters presented in contributions T1A1.5/92-112, T1A1.5/92-135, T1A1.5/92-138, and T1A1.5/93-032 and the video delay parameter presented in contribution T1A1.5/92-139 can be directly computed from the real-time measurement data. It is intended that the real-time system be used for the collection of video quality data quickly and efficiently in the laboratory or in the field.

This contribution describes the system from both the hardware and the software points of view, and presents implementation and timing information about the spatial and temporal measures. It concludes by comparing the results of the real-time system with those of our laboratory system as a means of validating the methods of measurement for the ITS video quality parameters. The validation results revealed that the gain and the frequency response of the measurement system are important for proper implementation of the ITS video quality measures.

This contribution supplies a more in-depth explanation of the real-time system demonstrated to T1A1.5 members at the April meeting in Boulder, CO. The real-time system conclusively demonstrates that the ITS video quality measures are practical and easily implemented in a field-deployable instrument. Potential uses for the real-time video quality system include processing of the T1A1.5 video tapes and live monitoring of the ITS video quality measures for objectively measuring the performance of video transmission systems.

2. System Description

The real-time system offers a compact and efficient way to collect video quality data. The system is configured to calculate the spatial and temporal measures for one video source. This source may be live video being input to a communication channel, or it may be taped video. For example, measurements could be taken using a subjective viewing tape or an HRC (hypothetical reference circuit) tape. The raw data is then stored to a file. Further data processing can be done to calculate quality parameters and predictions of subjective quality.

Sections 2.1 and 2.2 discuss the details of the hardware and software configurations respectively. Section 2.3 discusses the video sampling frequencies used by the real-time system.

2.1 Hardware Configuration

The primary component of the hardware consists of two image processing PC cards. Each card contains an on-board GSP (graphics signal processor) running at 30 MHz, a FPU (floating point unit), four Mbytes of VRAM (video random access memory), a cascaded ALU (arithmetic logic unit), a neighborhood processor, a digitizer, and display

hardware. All digitization of the video and image processing is done on the PC cards themselves. The results of the image processing (pixel sums, etc.) are passed to the host PC where they are stored in a file on a RAM (random access memory) disk for speed considerations. Figure 1 is a block diagram of the hardware configuration.

The PC cards are designed to digitize a luminance signal. Our input video is, therefore, the (luminance) Y channel of a component Betacam signal. The signal is routed through a video distribution amplifier to supply each board with a valid video signal.

Two image processing cards are required to simultaneously calculate both measures for a given input video signal. The software is written such that the temporal measure is calculated on PC Card 1, and the spatial measure is calculated on PC Card 2. Once the raw data has been calculated by the PC cards, it is transferred to the host PC via FIFO (first in first out) buffers which interface the PC cards to the host PC. The PC then writes the raw data to a binary file located on a RAM disk. It is faster to write to a RAM disk because the PC is writing to its main memory as opposed to writing to the hard drive, which has a greater average access time.

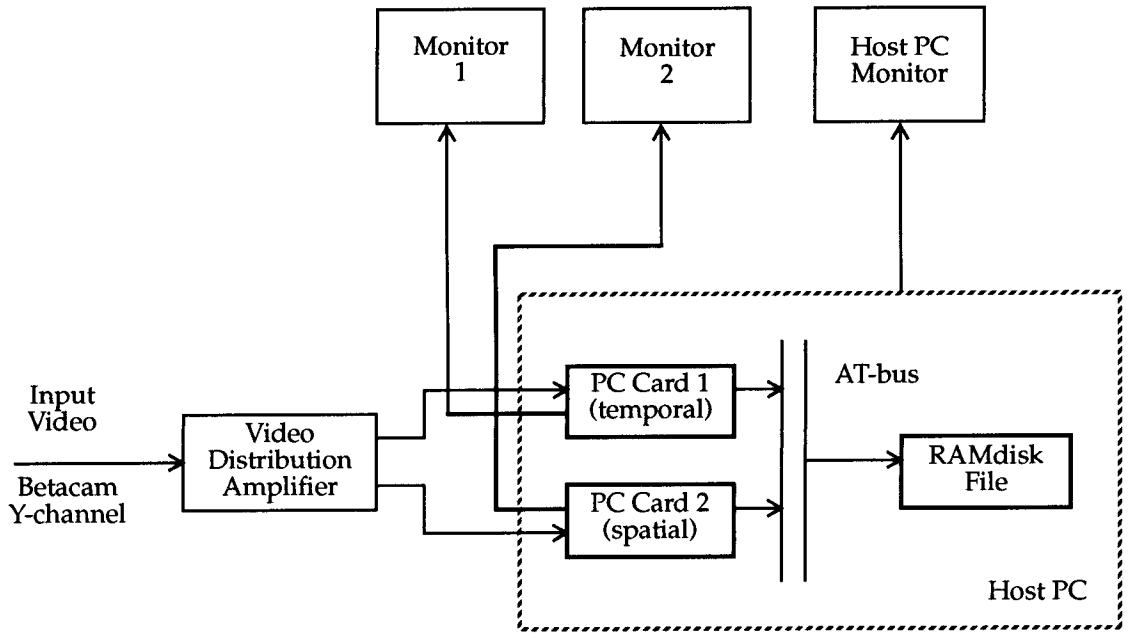

Figure 1 System overview: Hardware configuration.

2.2 Software Configuration

The software has a three-layer architecture as shown in Figure 2. The control program (written in the 'C' programming language) is designed to perform the required measurements. The program makes use of an extensive library of function calls which communicate information from the control program to the boards themselves. The board-level software (assembly code) is automatically downloaded from the PC upon power up. The assembly code interprets the op-codes passed to it from the control program and executes the requested board-level commands. The output of the requested

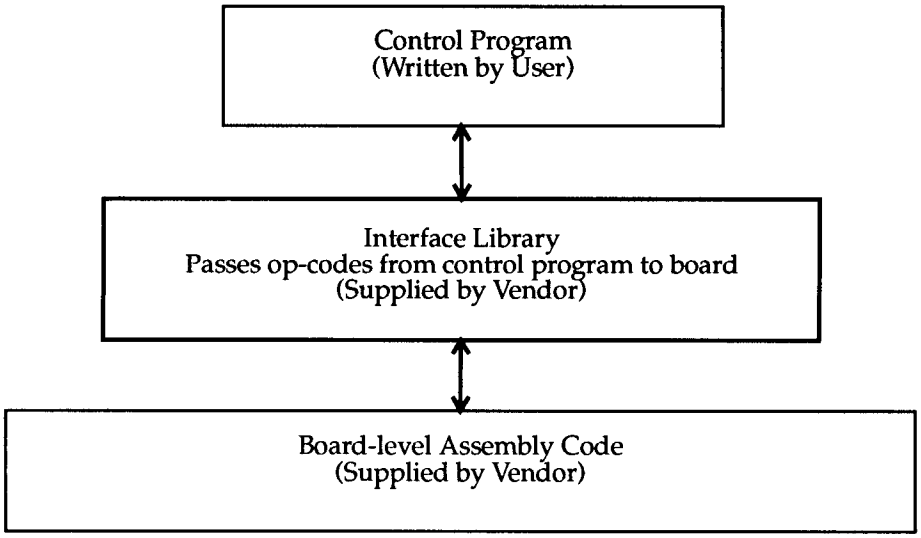

Figure 2 System overview: Software configuration.

commands can be anything from an image statistic to a displayable image. If a numeric result is requested, it is either stored in a Statistical LUT (lookup table) or in the FIFO buffer. The control program, through the use of a function call, can then retrieve data from the Statistical LUT or the FIFO buffer.

2.3 Digitization

The video sampling frequency for our PC cards is software selectable. The sampling rate f_s, can be chosen to be any frequency which, when multiplied by two, falls within the range of the fundamental frequency f_F, 25 - 30 MHz.

$$f_s = \frac{f_F}{2^n} \tag{1}$$

We chose to implement two sampling rates for the real-time system. The first sampling rate is the four-times subcarrier sampling frequency, $f_s = 4 \cdot f_{sc}$ where $f_{sc} = 3.5795$ MHz ($f_{sc} = 14.318$ MHz), for composite signals. The second rate chosen was the CCIR-601 component sampling frequency of $f_s = 13.5$ MHz . The sampling frequency is software-selectable on the command line used to execute the control program.

3. Parameter Implementation

The following sections will discuss the specific implementations of our spatial and temporal measures. There are some approximations which have been made to achieve

real-time operation. These will be pointed out in the following discussion. Also, the specifics of the process timing will be covered.

3.1 The Spatial Measure

The spatial measure is based upon the pseudo-sobel-filtered image. Several approximations to the pseudo-sobel filter have been made to accommodate the 8 and 16-bit arithmetic used by the real-time image processing boards. The original sampled image X is convolved (*) with the two kernels in equation (2). This results in two images Y_h and Y_v (see equation (3)). At this point, these intermediate results are signed 16-bit numbers. The valid data actually ranges over 10 bits. The first approximation employed is to right-shift the signed 16-bit data by two bits. This effectively divides the data by four. Next, the absolute value of each image ($|Y_h|$, $|Y_v|$) is calculated resulting in unsigned 16-bit numbers. The next approximation occurs when the hardware clips the unsigned 16-bit data to 255 if it exceeds 255. The clipping should rarely happen because of the previous right shifting of the data. The effect of the shift is taken into consideration when the final data is calculated.

$$h = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad v = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{2}$$

$$Y_h = h*X, \text{ and } Y_v = v*X \tag{3}$$

Once the data has been shifted and the absolute value has been taken, each image is in its unsigned 8-bit representation. The final filtered image, Y_{ps}, is the sum of the two images as shown in equation (4) where the symbol >> indicates a right shift. Summing two 8-bit images can result in a 9-bit value. When this occurs, our hardware overflows the data instead of clipping it. For example, a sum of 257 would result in a value of 1. Because this type of overflow is a rare occurrence, the overall results are not appreciably effected. The image Y_{ps} is displayed.

$$Y_{ps} = |Y_h \gg 2|_{255} + |Y_v \gg 2|_{255} \tag{4}$$

The spatial measure is the standard deviation of the image Y_{ps}. The sum of the pixels in Y_{ps} (S_1), the sum of the squares of pixels in Y_{ps} (S_2) and the number of pixels in Y_{ps} (N), given by equation (5), are used to calculate standard deviation. This data is stored to a file on the RAM disk. The scaling factors in the equations compensate for right shifting

the data by two bits.

$$S_1 = 4 \cdot \sum_{i=0}^{N-1} p_i$$
$$S_2 = 4^2 \cdot \sum_{i=0}^{N-1} p_i^2 \qquad (5)$$

N = Number of pixels in image Y_{ps} $\qquad$ p_i = i^{th} pixel in image Y_{ps}

The standard deviation can be calculated from the data using one of the equations given in equation (6).

$$\sigma = \sqrt{\frac{1}{N-1}(S_2) - \frac{N}{N-1}\left(\frac{S_1}{N}\right)^2}$$
$$\sigma \approx \sqrt{\frac{1}{N}(S_2) - \left(\frac{S_1}{N}\right)^2} \qquad (6)$$

3.1 .1 Spatial Measure Timing

The time required for each calculation discussed above is as follows:

- Grab Frame: 33.37 ms
- Convolve Image: 25.76 ms
- Calculate S_1 27.06 ms
- Map Y_{ps} through LUT to square pixel values: 24.80 ms
- Calculate S_2: 27.06 ms

Total: 138.05 ms

Thus, it takes a little over four frame times (where one frame requires 33.3 ms) to calculate the spatial measure. The fifth frame is the next frame grabbed. Thus, the spatial measure is calculated six times per second. Temporal subsampling of the spatial measure at rates less than 30 times per second has been shown to be acceptable in a prior contribution. Contribution T1A1.5/92-135 described results where the spatial measure was sampled at a rate of three times per second without affecting the subjective to objective correlation results.

3.2 The Temporal Measure

The temporal measure requires the subtraction of two consecutive frames. The calculation must therefore be performed in one frame time (33 ms) for real-time operation. To meet this constraint, only one field is grabbed and processed. This leaves

one half frame time (16.5 ms) to perform the necessary calculations. The calculations can be performed within 16.5 ms if the field is also subsampled by two in the horizontal direction. This spatial subsampling of the temporal measure does not appear to appreciably affect the overall temporal measure results. Every other subsampled field is displayed on the monitor. The other is stored in non-displayable memory.

Once two fields have been grabbed, the subtraction and absolute value is performed by mapping the two fields through the PC card's ALU (equation (7)). The result, Y_{fd}, is stored and displayed on the monitor.

$$\begin{gathered} Y_{fd} = |f_n - f_{n-2}| \\ Y_{fd} = \text{absolute field difference image} \\ f_n = \text{field at time n} \end{gathered} \tag{7}$$

The temporal measure is the mean of image Y_{fd}. The sum of the pixels in Y_{fd} (S_3) and the number of pixels in Y_{fd} (N) (see equation (8)) are used to calculate the mean. This data is stored to a data file on the RAM disk.

$$\begin{gathered} S_3 = \sum_{i=0}^{N-1} p_i \\ N = \text{Number of pixels in subsampled image } Y_{fd} \\ p_i = i^{th} \text{ pixel in image } Y_{fd} \end{gathered} \tag{8}$$

The mean can be calculated from the raw data using equation (9).

$$m = \frac{1}{N} \cdot S_3 \tag{9}$$

3.2 .1 Temporal Measure Timing

The time required for each calculation discussed above is as follows:

- Grab and subsample Field: 16.68 ms
- Calculate absolute field difference image Y_{fd}: 7.11 ms
- Calculate S_3: 7.96 ms

Total: 31.75 ms

By calculating the mean of the field difference instead of the standard deviation, the temporal measure can be calculated in under 33 ms. The mean appears to contain essentially the same quality information as the standard deviation (the original temporal measure developed in prior contributions used the standard deviation). Thus, the temporal measure is calculated 30 times per second.

4. Validation

The results obtained using the real-time system were validated by comparison with results obtained using our laboratory system which contained different digitizing and processing equipment. The reference tape made available to T1A1.5 was used as the video source (T1A1.5/92-157). The source was processed by both the real-time system and our laboratory system. The laboratory system was programmed to implement all of the software approximations in the real-time system. Thus, we were able to quantify the differences due to hardware of the two systems. The data collected from the laboratory and real-time systems were then compared to calculate the error between the two systems. Two main sources of error were discovered during this process. The first is the difference in the gain of the laboratory and real-time systems, and the second is the difference between the frequency responses of the two systems. Both of these issues will be discussed in the following sections.

4.1 System Gain

The frame grabbing boards in the laboratory system and the real-time system are calibrated to digitize 100 IRE to level 235 and 7.5 IRE to level 16 as specified in CCIR-601 recommendations. However, due to the setup of our video laboratory, the video follows different paths to reach the frame grabbers in our laboratory and real-time systems. Differences between the two paths include different length cables, a video switcher, and a component video patch panel. Hence, the video signals arriving at the frame grabbing boards in the real-time system and the laboratory system are slightly different. Appropriate gain compensation could be performed using either hardware or software. Using hardware, the digitizing board could be recalibrated to perform the gain compensation. However, this adjustment would only be acceptable for a given configuration of the laboratory. The software solution, and the one that we adopted, is to calculate the gain corrections for the laboratory and the real-time systems and apply it to the data that they produce. The gain correction can be calculated using the white and 7.5 IRE black levels in the SMPTE color bars as follows. Calculate the average white level for the laboratory system and the real-time system, W_{lab} and W_{rts} respectively. Calculate the average 7.5 IRE black level for the laboratory system and the real-time system, B_{lab} and B_{rts} respectively. Calculate the gain factors G_{lab} and G_{rts} using equation (10).

$$G_{lab} = \frac{W_{lab} - B_{lab}}{235 - 16} \qquad G_{rts} = \frac{W_{rts} - B_{rts}}{235 - 16} \tag{10}$$

The correction can be made by dividing the data obtained with each system by the respective gain in equation (10). Our validation run resulted in a laboratory system gain of approximately 0.99 and a real-time system gain of approximately 0.95. ITS recommends that the gain be quantified for any system implementing the ITS quality metrics.

If we define the error between the laboratory data d_{lab} and the real-time data d_{rts} as

follows

$$Error(i) = |d_{lab}(i) - d_{rts}(i)|, \tag{11}$$

we can calculate the percentage root mean square error relative to the laboratory system as

$$\%\ \text{Relative RMSE}\ (d_{lab}) = \frac{RMS(Error)}{RMS(d_{lab})} \cdot 100. \tag{12}$$

Using only gain-compensated data, the error between the laboratory system and the real-time system for the statistics calculated on Y_{fd} and Y_{ps} are

$$\%\ \text{Relative RMSE}\ (\text{mean}\ (Y_{fd}(i))) = 2.12\%$$

$$\%\ \text{Relative RMSE}\ (\text{standard deviation}\ (Y_{ps}(i))) = 5.03\%.$$

The gain-compensated laboratory system and real-time system data can be seen in Figure 3 (absolute field difference) and Figure 4 (pseudo sobel). Note that the pseudo sobel is only calculated 6 times/sec, thus there are only 52 samples relative to the 260 samples of the absolute field difference.

4.2 Frequency Response

In light of the five percent error in the pseudo-sobel measure, we looked for other sources of error. In doing so, we discovered that the frame grabber in our laboratory system has a much flatter frequency response than the frame grabber in our real-time system. This is due to the presence of a 5.3 MHz (attenuated $-5\text{dB} \pm 2\text{dB}$) pre-filter in the real-time system.

We injected a multiburst signal into our real-time system board to find a rough approximation of its frequency response. The multiburst signal was generated using a signal generator, and its frequencies are 0.5 MHz, 1.5 MHz, 2.5 MHz, 3.58 MHz, 4.1 MHz, and 4.5 MHz. We then calculated the magnitude of the response at each frequency using the standard deviation of the signal within the portion of the image containing the given frequency. Our experimental results showed that the real-time system pre-filter was attenuated -1.8 dB at 4.5 MHz.

Using this rough frequency response, we were able to calculate the time-domain filter equivalent of the real-time system using a signal processing program. This filter was then applied to each line of video in each image digitized by our laboratory system to simulate the real-time pre-filter. The filtered images were then processed as usual. The percent relative root mean square error was again calculated using both gain compensation and rough frequency compensation. The results are

$$\%\ \text{Relative RMSE}\ (\text{mean}\ (Y_{fd}(i))) = 1.54\%$$

$$\%\ \text{Relative RMSE}\ (\text{standard deviation}\ (Y_{ps}(i))) = 1.00\%.$$

The data with additional compensation can be seen in Figure 5 (absolute field difference) and Figure 6 (pseudo sobel). The frequency compensation accounts for a large portion of the error, especially in the pseudo-sobel measure. This is because the

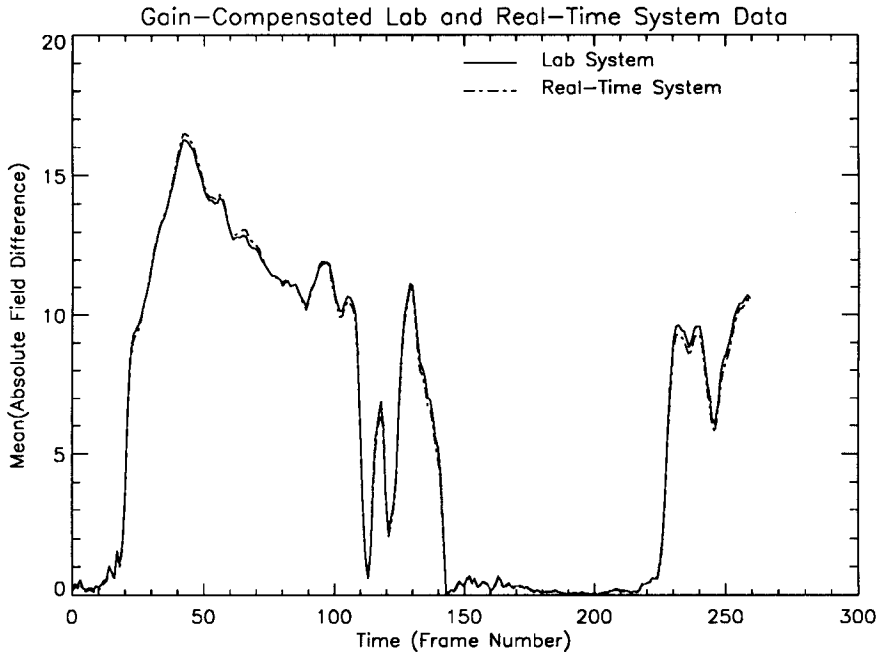

Figure 3 Absolute field difference, gain-compensated laboratory system and real-time system data.

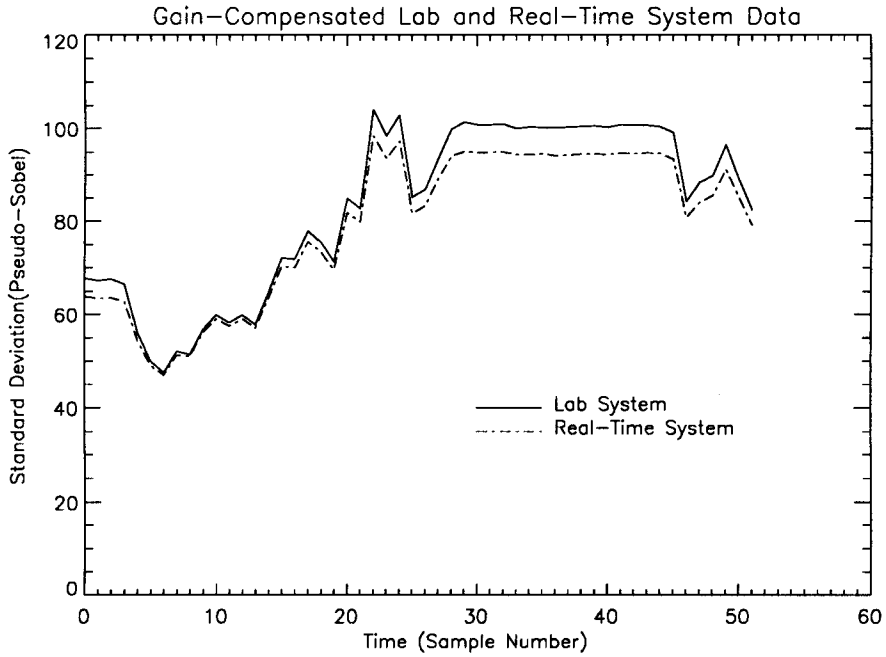

Figure 4 Pseudo sobel, gain-compensated laboratory system and real-time system data.

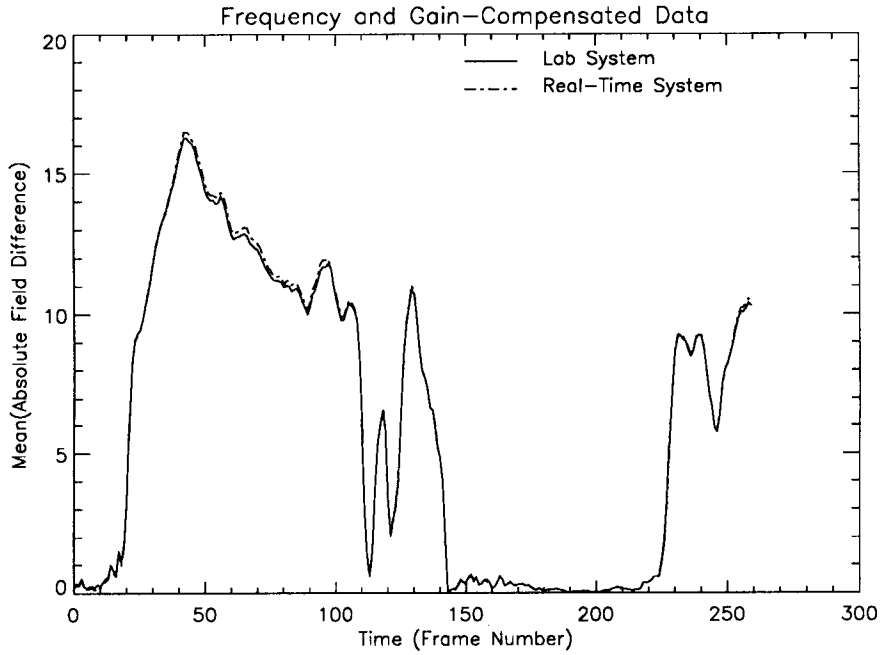

Figure 5 Absolute field difference, frequency-compensated and gain-compensated data.

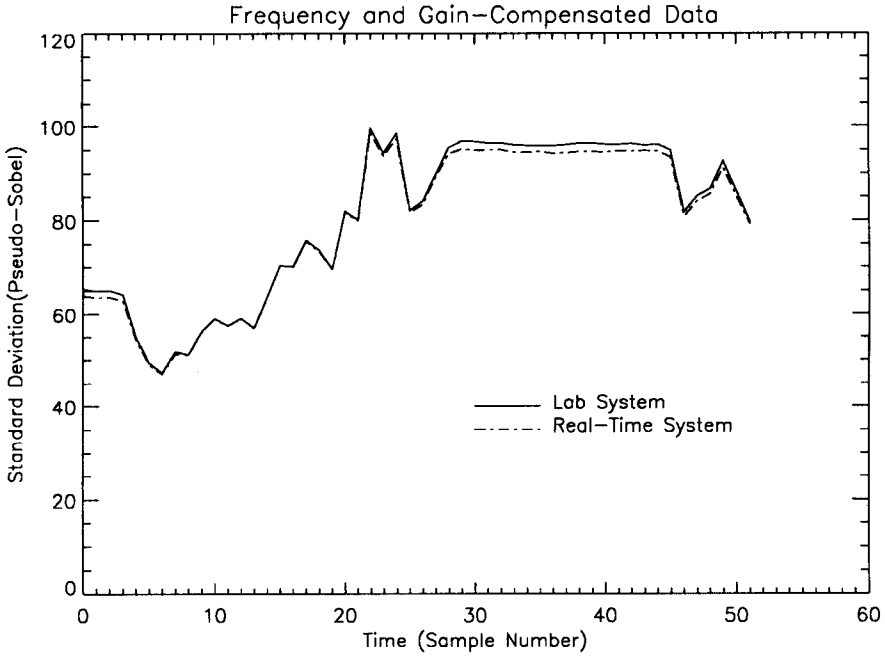

Figure 6 Pseudo sobel, frequency-compensated and gain-compensated data.

pseudo-sobel is sensitive to spatial frequencies. If the high-frequency content of the image has been filtered out, the energy in the pseudo-sobel image will decrease accordingly. The spatial measure was, in effect, detecting differences in the spatial resolution of the laboratory and real-time system frame-grabbing boards. Thus, the spatial measure was performing as designed.

Once the video signal has been filtered, it is impossible to compensate for the lost information. CCIR-601 recommends a flat frequency response with a ripple of no more than ±0.05 dB out to 5.75 MHz. Therefore, the best solution for the real-time system may be a hardware solution whereby the low-pass filter on the frame-grabbing board is modified such that the cutoff frequency is moved to a higher frequency. This problem is mentioned because it can contribute a significant amount of error. ITS advises that the flatness of the frequency response be tested for any system implementing the ITS quality metrics.

The error may be reduced further if a more accurate filter corresponding to the difference between the frequency responses of the laboratory system digitizer and the real-time system digitizer can be modeled.

5. Conclusion

We have demonstrated that the ITS video quality measures are practical and easily implemented in a PC-based, real-time, field-deployable instrument. The measures are also repeatable in that two different systems give the same objective numbers.

To assure this repeatability, this contribution has shown that any laboratory implementing the ITS video quality metrics should definitely quantify the following attributes of their system:

- system gain
- system frequency response

Without knowing this important information, it will be difficult to properly implement the measures.

Planned uses for the real-time system include processing the T1A1.5 HRC (hypothetical reference circuit) and subjective viewing tapes. Potential future uses include automatic end-to-end monitoring of the ITS video quality measures for objectively measuring the performance of digital video transmission systems. This will be made possible by adding the ability to calculate video delay and by including programs which allow one measurement to be made simultaneously on two video sources, for example source and destination video.