COMMITTEE T1
CONTRIBUTION

Document Number: T1Q1.5/90-114

****

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

****

TITLE: MISSING FRAME RATIO (MFR), A MEASURE OF JERKINESS

****

ISSUE ADDRESSED: A TECHNIQUE FOR MEASURING THE NUMBER OF MISSING FRAMES IN VTC/VT OUTPUT VIDEO

****

SOURCE: NATIONAL TELECOMMUNICATIONS AND INFORMATION ADMINISTRATION
INSTITUTE FOR TELECOMMUNICATION SCIENCES
(Stephen Wolf)
(Keith Junker)

****

DATE: April 24, 1990

****

DISTRIBUTION TO: T1Q1.5

****

KEYWORDS: VIDEO TELECONFERENCING, VIDEO TELEPHONY,
ARTIFACTS, JERKINESS, PERFORMANCE ASSESSMENT

****

Introduction

The omission of fields and/or frames by the codec before transmission followed by field and/or frame repetition by the codec on the receiving end contributes to motion jerkiness in VTC/VT systems. Reference (1) has recommended that Transmitted Frame Rate (TFR) be used as a measure of this frame omission by VTC/VT codecs. This contribution proposes a related measure that has been used by ITS to measure the ratio of missing frames in the output of a VTC/VT codec. The method of measurement for missing frame ratio (MFR) presented here is general and applicable to the wide range of video coding schemes in use today. Other advantages include

Application to any test scene

Since MFR could (and usually does) vary automatically depending upon the scene, it is extremely important that the measurement technique be applicable to arbitrary scenes. This requirement is consistent with reference (2) that proposes different performance standards depending upon the application. Here, jerkiness and spatial resolution are traded against one another to achieve performance that is optimized for a particular application, or set of test scenes. Since MFR can vary significantly depending upon the test scene, measurements based on injected test patterns or waveforms may yield irrelevant results. The method of measurement for MFR presented here can be used for any test scene.

Dynamic computation

As previously mentioned, MFR may vary dynamically within a test scene. The method of measurement for MFR presented here can also be performed dynamically.

Application to codecs that perform field and/or frame interpolation

Sophisticated codecs may omit fields and/or frames on transmit and perform field and/or frame interpolation on receive. There already exists an NTSC chip set that performs real time motion interpolation when converting 30 frames per second (interlaced) to 60 frames per second (non-interlaced).[3] A sophisticated codec that performs a good job interpolating frames on the receiver end should not be penalized because it has a lower Transmitted Frame Rate (TFR). The measurement method for MFR presented here does not penalize codecs that perform field and/or frame interpolation on receive, provided they accurately perform the interpolation.

The MFR measurement technique assumes that some motion or changing scenery is present in the video. This does not present a problem since computation of MFR for completely static video scenes is not an issue. Other methods may be required if it is desired to measure jerkiness not resulting from field and/or frame omission. Several methods of measuring these other components of jerkiness have already been proposed in reference (4).

Computation of Missing Frame Ratio

The technique for computing missing frame ratio is conceptually simple and easily implemented. MFR is computed using the digitized input and codec output video images. The steps are as follows

1. Computation of closest matching input video frame

   The best matching input frame (for a given codec output frame) is found by computing the error difference images between the selected output frame and all reasonable input frames. Assuming that some motion or changing scenery is present in the input video, the variance of the error (accumulated over all pixels in the error image) will be minimized for the best aligned input video frame. The reader is referred to equation 1 of the Appendix for a mathematical definition of the process of computing the closest input video frame.

2. Repetition of step 1 above for each codec output frame

   Step 1 above is performed for each codec output frame. The missing frame ratio is then readily computed as the number of input frames that are missing in the codec output (i.e., the number of input frames that did not match any of the codec output frames), divided by the total number of video frames. The reader is referred to equation 2 of the Appendix for a mathematical definition of the process of computing MFR. The Appendix also contains information on how to perform steps 1 and 2 in a computationally efficient manner.

Figures 1 and 2 illustrate the computation of MFR applied to a video scene that contained motion. The top row of Figure 1 shows four consecutive frames that were captured every 1/30 sec, left to right, from the original NTSC video scene. This original NTSC video scene was injected into a VTC/VT coder/decoder (codec) running at 1/4 the digital signal one (DS1) rate of 1.544 Mbps. The codec output is shown in the bottom row of Figure 1. The solid lines in Figure 1 show the pairing of the input video frames with the codec output frames as found from steps 1 and 2 above. Figure 2 shows the error difference images (input frame minus codec output frame) that were used to determine the best matching input frame for each codec output frame. In Figure 2, white and black are positive and negative error, respectively, while the gray background represents no error. The top row in Figure 2 shows the error difference images between the four input frames (top row of Figure 1) and the first codec output frame (bottom, left image in Figure 1). Of the four error images in the top row of Figure 2, the first one (leftmost) contains the smallest error (least amount of black and white). Thus, codec output frame 1 is paired with input frame 1 in Figure 1. Rows two, three, and four of Figure 2 give the corresponding error difference images for the second, third, and forth codec output frames in Figure 1. Clearly, the particular codec tested discarded every other NTSC input video frame and performed frame repetition on the output to fill in for the missing video frames. MFR for this example is calculated as two divided by four (or .5), since two of the four input video frames were missing in the output.

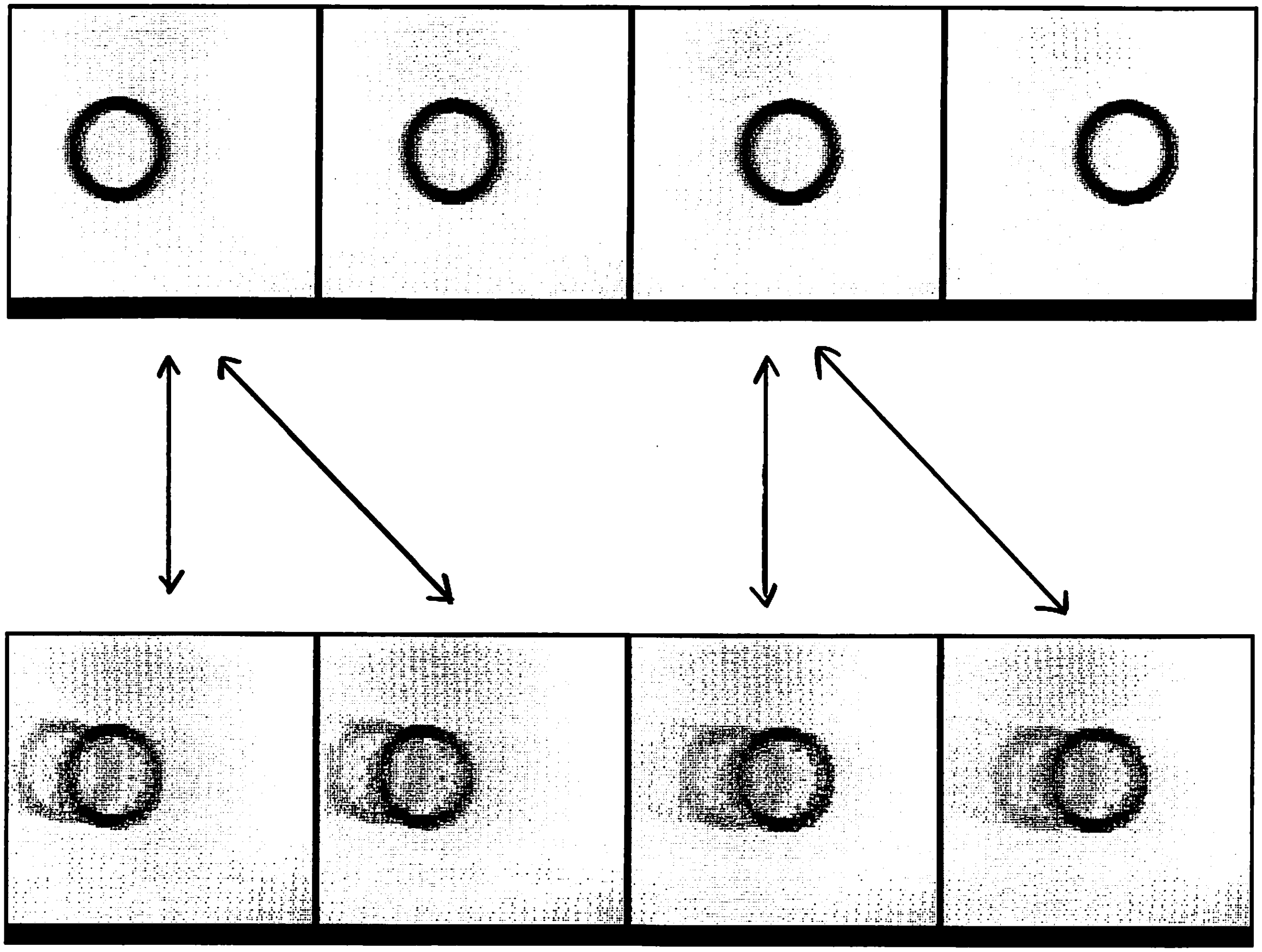

Figure 1. Top row – original NTSC grabbed every 1/30 sec from left to right. Bottom row – VTC/VT codec output. Solid lines represent the closest input video frame for each codec output frame.

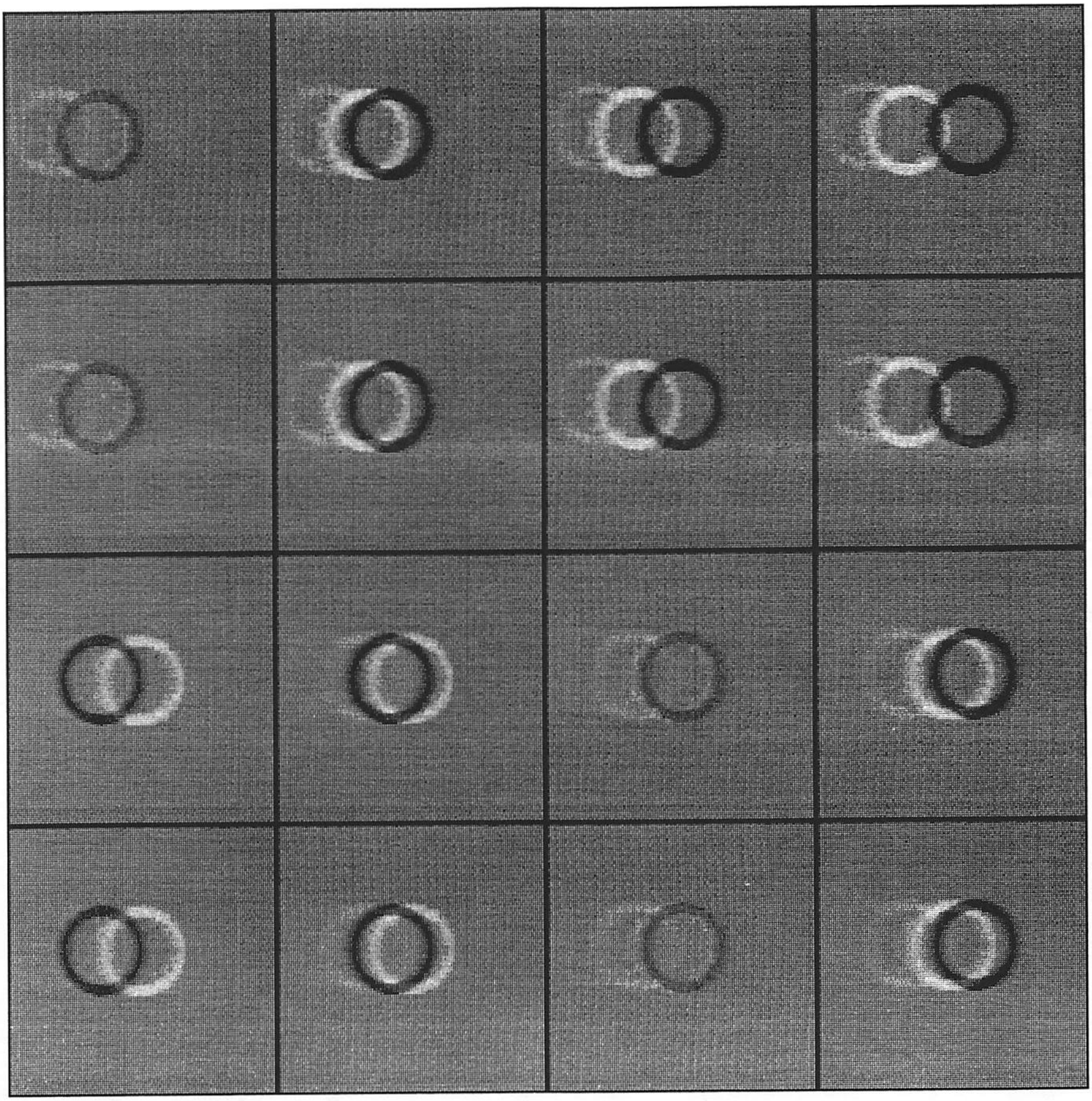

Figure 2. Error difference images (input–output) of Figure 1. Top row – NTSC input (top row in Figure 2) minus codec output frame 1 (bottom row, leftmost frame in Figure 2). Second, third, and forth rows are NTSC input minus codec output frames 2, 3, and 4, respectively.

REFERENCES

(1) Delta Information Systems, "Transmitted Frame Rate", Committee T1 contribution T1Q1.5/90-109, Jan 30, 1990.

(2) Institute for Telecommunication Sciences, "Definitions of Video Terms, Application Groups, and Levels of Performance for VTC/VT", Committee T1 contribution T1Q1.5/90-112, April 24, 1990.

(3) Wilson, A., "Six-Chip Set Rids Imaging of the Flickers", Electronic System Design Magazine, Dec., 1988, pp. 28-29

(4) Institute for Telecommunication Sciences, "Feature Extraction for Automated Quality Assessment of Digitally Transmitted Video", Committee T1 contribution T1Q1.5/90-108, Jan 30, 1990.

APPENDIX: EQUATIONS

This appendix describes the equations that were used to compute the missing frame ratio (MFR) parameter.

Equation (1): Computing Closest Input Frame

Let $i(v,h,t_i)$ be the digitized input video sequence where v is the vertical sampling index, h is the horizontal sampling index, and t_i is the input frame sampling index. Here $v = \{1, 2, \ldots, N_v\}$, and $h = \{1, 2, \ldots, N_h\}$, where N_v is the total number of vertical pixels, and N_h is the total number of horizontal pixels. Similarly, let the digitized output video sequence be represented by $o(v,h,t_o)$, where t_o is the output frame sampling index. Assume that the input and output video sequences are sampled at the same frame rate. Then, given an output reference frame $t_o = r$, the temporal alignment problem is to find the closest corresponding input frame $t_i = m$. The temporal alignment proposed here assumes that *a priori* knowledge is available which gives the range of the closest corresponding input frame index (say, from $t_i = t_l$ to t_u, where t_l and t_u are the lower and upper limits, respectively) and that the input video sequence contained moving and/or changing scenes. Then the closest matching input frame $t_i = m$ can be found as the t_i that minimizes the variance of the error (accumulated over all pixels in the error image) or

$$\left\{\frac{1}{N_v N_h}\sum_{v=1}^{N_v}\sum_{h=1}^{N_h}[i(v,h,t_i)-o(v,h,r)]^2\right\} - \left\{\frac{1}{N_v N_h}\sum_{v=1}^{N_v}\sum_{h=1}^{N_h}[i(v,h,t_i)-o(v,h,r)]\right\}^2$$

where t_i falls within the range from t_l to t_u, inclusive. A significant reduction in the number of computations could be obtained if a time code or other timing data was "burned" into every input video frame. Then the above summations would only have to be performed over the sub-regional area that contained the time code. The disadvantage of using a time code is that the response of the codec to a changing (frame by frame) time code could differ from the response of the codec to a test scene.

Equation (2): Missing Frame Ratio (MFR)

The temporal alignment given in equation (1) above is applied to every output video frame $o(v,h,t_o)$, where $t_o = \{1, 2, \ldots, N_o\}$, and N_o is the total number of output video frames in the test scene. The computation of the closest input video frame to output video frame $t_o = 1$, may be used to refine the estimates of the lower (t_l) and upper (t_u) input frame limits for output video frame $t_o = 2$. While performing the alignment, the closest input video frame index to each one of the output video frames is stored. Let the number of unique input frame indices within the set of stored indices be N_u. N_u will be less than N_o if input video frames have been omitted in the output. Then, the missing frame ratio (MFR) is calculated as

$$MFR = \frac{N_o - N_u}{N_o}$$