NTIA Report 90-264

Features for Automated Quality Assessment of Digitally Transmitted Video

Stephen Wolf

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary

Janice Obuchowski, Assistant Secretary
for Communications and Information

June 1990

PREFACE

The National Telecommunications and Information Administration (NTIA) has funded the Institute for Telecommunication Sciences (ITS) to develop technology independent measures of video quality performance for application to modern transmission systems. Such modern transmission systems include video teleconferencing/video telephony (VTC/VT), digital television, wideband integrated services digital networks (ISDN), high resolution graphics transmission, and high definition television (HDTV).

This report summarizes several of the technology independent measures of video quality that have been developed for the ongoing ITS video quality performance assessment project.

The views, opinions, and findings contained in this report are those of the author only. The report does not reflect NTIA, ITS, or any other agency position, policy, or decision unless so designated by other official documentation.

TABLE OF CONTENTS

	Page
PREFACE	iii
TABLE OF CONTENTS	v
LIST OF FIGURES	vii
LIST OF TABLES	ix
LIST OF ACRONYMS	x
ABSTRACT	1
1. INTRODUCTION	1
1.1 Background	2
1.2 Automated Video Quality Measurement System Overview	4
2. DESCRIPTION OF FEATURES	6
2.1 Common Video Compression Artifacts	6
2.2 Desirable Properties of Features	9
2.3 Alignment Of Original And Distorted Video Imagery	11
2.3.1 Single-frame Temporal Alignment	11
2.3.2 Multi-frame Temporal Alignment	13
2.4 Preconditioning Of The Sampled Video	17
2.5 Spatial Blurring Features	18
2.5.1 Feature Extraction Technique	18
2.5.2 Sample VTC/VT Results	22
2.6 Blocking, Edge Busyness, and Image Persistence Features	32
2.6.1 Feature Extraction Technique	33
2.6.2 Sample VTC/VT Results	37
2.7 Jerkiness Feature Using Position Errors	42
2.7.1 Feature Extraction Technique	43
2.7.2 Sample VTC/VT Results	46

2.8 Jerkiness Feature Using Difference Image 59
2.8.1 Feature Extraction Technique 60
2.8.2 Sample VTC/VT Results 62

3. CONCLUSIONS AND RECOMMENDATIONS 66

4. ACKNOWLEDGEMENTS 68

5. REFERENCES 68

6. BIBLIOGRAPHY 71

7. APPENDIX A: EQUATIONS 72

8. APPENDIX B: FILTERS 80

LIST OF FIGURES

		Page
Figure 1.	Automated video quality measurement system.	5
Figure 2.	Single-frame and multi-frame alignment.	15
Figure 3.	Error difference images (input-output) of Figure 2.	16
Figure 4.	Camera interlace effects caused by horizontal motion.	20
Figure 5.	VTC/VT imagery containing rotational motion. . . .	23
Figure 6.	Sobel filtered edge extracted VTC/VT imagery of Figure 5.	25
Figure 7.	VTC/VT imagery containing upper body motion. . . .	27
Figure 8.	Leftmost column of Figure 7 expanded.	28
Figure 9.	Sobel filtered edge extracted VTC/VT imagery of Figure 7.	30
Figure 10.	Sobel filtered edge extracted VTC/VT imagery of Figure 8.	31
Figure 11.	VTC/VT imagery of moving black ring against white background.	38
Figure 12.	Sobel difference image of Figure 11.	39
Figure 13.	Sobel difference imagery of Figure 7.	41
Figure 14.	Four sequential VTC/VT images for a horizontally moving ball.	48
Figure 15.	Four sequential VTC/VT images for a diagonally moving ball.	50
Figure 16.	Positions of moving ball as a function of field number for fast motion at the horizontal angle.	52
Figure 17.	Positions of moving ball as a function of field number for fast motion at the diagonal angle.	53
Figure 18.	The aligned motion paths for the diagonal case in Figure 17.	54
Figure 19.	TRMS-PE plotted as a function of horizontal ball speed for code rates of 1/4 DS1 and DS1	57
Figure 20.	TRMS-PE plotted as a function of code rate for the fast speed group and diagonal motion.	58
Figure 21.	Difference images for VTC/VT imagery of Figure 7.	63

Figure 22. Time history of SD-DI for the difference images of Figure 21. 65

LIST OF TABLES

Page

Table 1. Common Video Compression Artifacts 8
Table 2. Spatial Blurring Features For VTC/VT Imagery Of Figure 6 26
Table 3. Spatial Blurring Features For VTC/VT Imagery Of Figure 9 32
Table 4. Spatial Blurring Features For Figure 12 40
Table 5. False Edge Features For Figure 12 40
Table 6. Spatial Blurring Features For Figure 13 42
Table 7. False Edge Features For Figure 13 42
Table 8. Summary Of TRMS-PE Results 56
Table 9. Summary Of SD-DI Features For Figure 22 66

LIST OF ACRONYMS

D

DS1	Digital Signal 1 (1.544 Mbps)

F

FFT	Fast Fourier Transform

H

HDTV	High Definition Television

I

ISDN	Integrated Services Digital Networks

M

MFR	Missing Frame Ratio
M-NSDI	Mean of Negative Sobel Difference Image
M-PSDI	Mean of Positive Sobel Difference Image
M-SI	Mean of Sobel Image

N

NPLT-NSDI	Number of Pixels Less Than Threshold of Negative Sobel Difference Image
NPGT-PSDI	Number of Pixels Greater Than Threshold of Positive Sobel Difference Image
NPGT-SI	Number of Pixels Greater Than Threshold of Sobel Image
NTSC	National Television Systems Committee

R

RMS-NSDI	Root Mean Square of Negative Sobel Difference Image
RMS-PSDI	Root Mean Square of Positive Sobel Difference Image

RMS-SI Root Mean Square of Sobel Image

S

SD-DI Standard Deviation of Difference Image
SD-NSDI Standard Deviation of Negative Sobel Difference Image
SD-PSDI Standard Deviation of Positive Sobel Difference Image
SD-SI Standard Deviation of Sobel Image

T

TM-SD-DI Temporal Mean of Standard Deviation of Difference Image
TRMS-PE Temporal Root Mean Square Position Error
TRMS-SD-DI Temporal Root Mean Square of Standard Deviation of Difference Image
TSD-SD-DI Temporal Standard Deviation of Standard Deviation of Difference Image

V

VTC/VT Video Teleconferencing/Video Telephony

FEATURES FOR AUTOMATED QUALITY ASSESSMENT OF DIGITALLY TRANSMITTED VIDEO

Stephen Wolf*

ABSTRACT

This report describes an automated method of video quality assessment based on extraction and classification of features from sampled input and output video. The first subsystem of the automated video quality measurement system is the feature extraction subsystem. Features are extracted from the sampled video that quantify many of the distortions present in modern digital compression and transmission systems. The feature measurements may then be injected into a quality classification subsystem which will determine the overall quality rating of the video. This report discusses the first subsystem of the automated video quality assessment system, namely the feature extraction subsystem. The measurement techniques used to extract a number of useful features are discussed in detail. Results are presented using sampled video teleconferencing data that contained common video compression artifacts.

Key words: American National Standards; ANSI; feature extraction; image processing; video quality; video teleconferencing

1. INTRODUCTION

As the world prepares to enter the age of digitally transmitted video services such as video teleconferencing/video telephony (VTC/VT), digital television, wideband integrated services digital networks (ISDN), high resolution graphics transmission, and high definition television (HDTV), new quality assessment techniques are needed. Traditional techniques for estimating video quality degradation during transmission have been based on analog measures of the transmission signal. These parameters are not adequate for assessing video quality when images are impaired by the many new types of distortions introduced by the modern digital transmission systems given above. In such cases, the video transmission quality is often a function of the type of imagery being transmitted (line drawings, natural scenes, etc.). Since the information normally has been compressed, small transmission errors due to channel impairments can have significant effects on the received video quality. As a result, viewing panels have been used to evaluate these modern

* The author is with the Institute for Telecommunication Sciences, National Telecommunications and Information Administration, U.S. Department of Commerce, Boulder, CO 80303.

distortion effects on video quality. Unfortunately, this approach is time consuming, expensive, and requires special care to prevent wide variations between tests. CCIR Recommendation 500-3 (1986) and Report 405-5 (1986) discuss in detail the methodology for conducting subjective assessment of the quality of television pictures.

New, objective measures of video transmission quality are needed by standards organizations, end users, and providers of advanced video services. Benefits would include impartial, reliable, repeatable, and cost effective measures of video and image transmission system performance and increased competition among providers as well as a better capability of procurers and standards organizations to specify and evaluate new systems.

1.1 Background

Extensive studies have been performed in recent years regarding quality assessment of video pictures. Most of the work falls into one of the following three groups:

1. Subjective quality assessment of still pictures or motion video.

2. Objective quality assessment of video components or systems based on output responses to injected test waveforms or patterns. The objective measurements are sometimes modified to account for characteristics of the human visual system.

3. Objective quality assessment of still pictures or motion video based on extraction of features directly from the video picture. The original (undistorted) picture is usually available for comparison. Since digital sampling of the video is performed, the objective measurements are sometimes modified to account for the effects of the video display device. In addition, characteristics of the human visual system are sometimes incorporated so that the objective measurements correspond more closely to the subjective rating.

CCIR Report 313-6 (1986) provides an extensive bibliography regarding assessment of the quality of television pictures. Nearly all of

publications listed in the report deal with the subjective quality assessment described in group (1) above. CCIR Recommendation 567-2 (1986) describes a set of objective measures which fall into group (2) above. CCIR Recommendation 654 (1986) defines relationships between the objective measurements and subjective picture quality, assuming that only one distortion type is present. The works of Biberman (1973), Higgins (1977), Task (1978, 1979), Carlson and Cohen (1980), and Barten (1987, 1988) also fall into group (2) above, since the quality measures are a function of the frequency responses (test waveforms are sinusoids) of the video and human vision systems. Meiseles (1988) has proposed a measurement of dynamic resolution based on rotating test patterns. Group (3) above includes the work of Mannos and Sakrison (1974), Sakrison (1977), Limb (1979), Pearson (1980), Toit and Lourens (1988), Ohtsuka et al. (1988), Miyahara (1988), and Tomich et al. (1989). Here, quality measures are normally developed as a weighted error of the distorted image relative to the original image.

The objective techniques of group (3) above are most applicable to video scenes which have undergone digital compression and transmission. Performance of image compression algorithms are a function of the type of imagery which is being compressed. A compression algorithm designed to perform well on one type of imagery, say natural scenes, may perform poorly on another type of imagery, like line drawings. In addition, the effects of transmission channel impairments (such as bit errors) must be determined by examining the resultant decoded or uncompressed image. Thus, video quality measurements based upon injected test signals, such as the techniques in group (2), could yield objective quality ratings that differ substantially from the subjective quality ratings. For an overview of image data compression, the reader is referred to Nesenbergs (1989).

Very little work in group (3) has been performed on video scenes that contain motion. Even recent papers which propose techniques in group (3) for motion video (Miyahara, 1988, Ohtsuka et al., 1988) do not evaluate their techniques using motion video. In practice, alignment of undistorted video and distorted video (from a wide class of video compression systems) requires careful consideration. Automated techniques for performing proper alignment of undistorted and distorted video will be discussed in detail later in this report.

1.2 Automated Video Quality Measurement System Overview

This report discusses a method for objectively measuring video quality based on feature extraction from digitized video imagery and classification techniques. Figure 1 gives an overview of the automated video quality measurement system. The computer-based approach extracts objective video quality features directly from captured video images. Video quality features extracted from the sampled imagery are chosen to be sensitive to user applications, video compression artifacts, and the effects of modern transmission channel impairments. In this report, a candidate set of features that quantify the presence of video compression artifacts has been developed by the author and his associates. Certain desirable properties of features, to be covered later in this report, guided this initial feature development and selection process. The objectively measured features are interpreted by a quality classification system to produce an overall quality rating comparable to that provided subjectively by a panel of viewers. Subjectively rated video data and psychological results from studies on human perception of video quality are used to assist in the design of the feature extraction and quality classification subsystems. In addition, not shown in Figure 1, certain a priori knowledge may be input to the feature extraction and quality classification subsystems to improve their performance. Examples of a priori control parameters include characterization of the display device which will be used to view the video, the viewing distance, or the type of video service.

The primary goal of the approach is to obtain an objective assessment of video quality that emulates the subjective rating. The goal is accomplished by selecting a set of features measured from the video imagery which correlates well with artifacts noticeable to the viewer, and by incorporating statistical and psychological results obtained from subjective evaluation of video imagery. The candidate set of features will be extracted from subjectively rated video imagery that exhibits a wide range of distortions. Then pattern recognition and classification techniques will be applied to determine the mapping of these objectively measured features into subjective quality space (as in Figure 1). Through application of pattern recognition and classification techniques, some of the features in the candidate set may prove to be redundant or ineffective in determining video quality. Hence, these

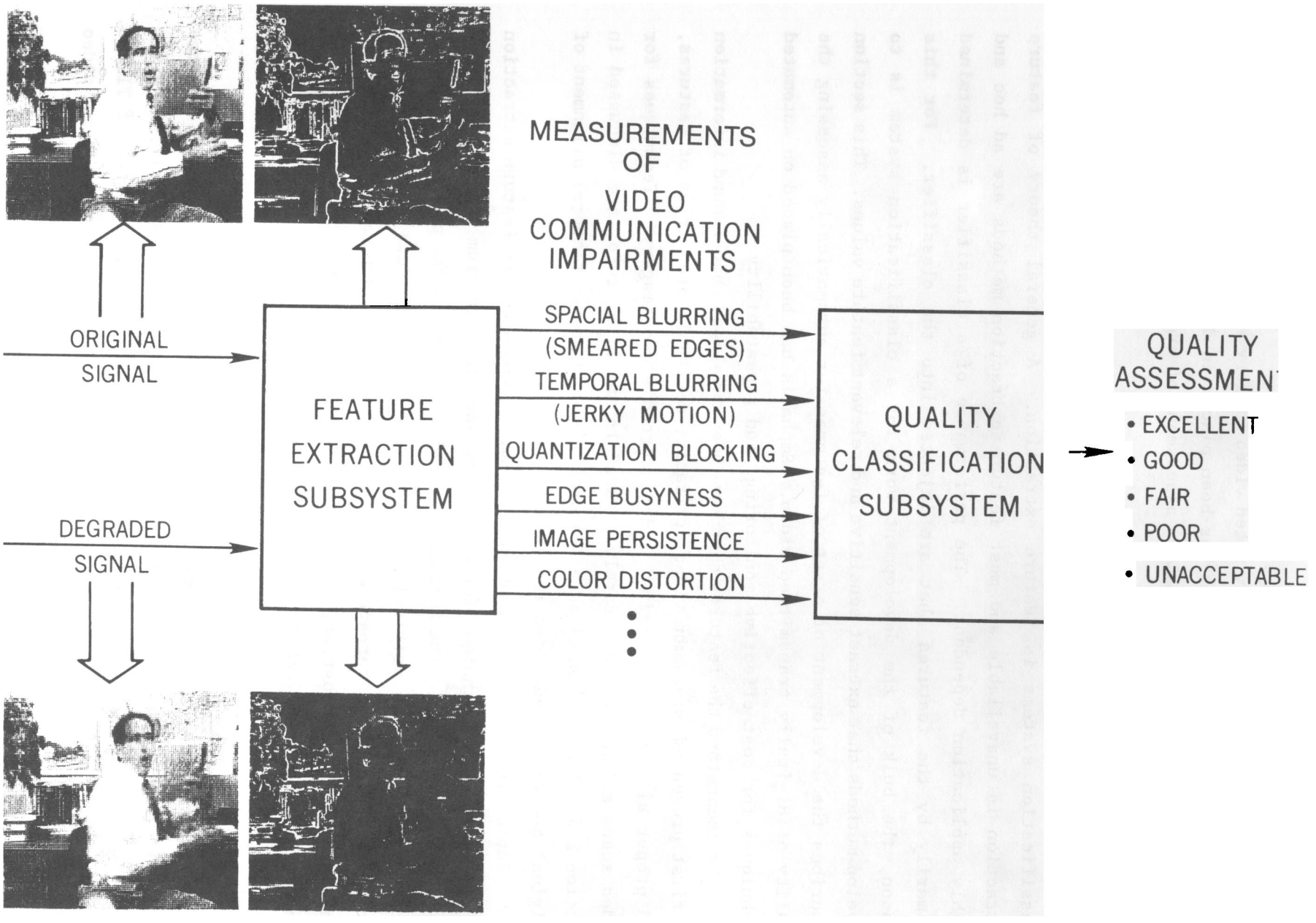

Figure 1. Automated video quality measurement system.

redundant or ineffective features may be discarded. Since subjectively rated video imagery was unavailable at the time of writing this report, emphasis has been placed on development of a candidate set of features for automated quality assessment of digitally transmitted video.

2. DESCRIPTION OF FEATURES

The most difficult process in virtually all pattern recognition and classification systems is feature extraction. A general theory of feature extraction is unavailable and most feature extraction methods are ad hoc and highly application dependent. The performance of a classifier is determined primarily by the features that are injected into the classifier. For this reason, the bulk of the development work for a classification system is to develop methods that extract sensitive and relevant feature values. This section describes the development of a set of features for automatically assessing the quality of digitally transmitted video. Emphasis has been placed on automated techniques for cost effective monitoring, and repeatability.

To understand the features that have been developed, background information is first presented on common video artifacts, desirable properties of features, and proper alignment of original and distorted video imagery. Techniques for video scene alignment, very rarely covered in the literature, are discussed in section 2.3. Calculation of some features requires proper temporal alignment of original and distorted video imagery.

Rationale for preconditioning the sampled video before feature extraction is discussed. The technique for extracting each feature from the sampled video is described in detail. The features objectively quantify the presence of common video artifacts. Of critical concern here is the computational time of a particular feature. Alternate algorithms are presented that reduce this cost of computation. For illustrative purposes, each feature extraction technique is demonstrated using VTC/VT data.

2.1 Common Video Compression Artifacts

The American National Standards Institute, Accredited Standards Committee T1, Working Group T1Q1.5 is drafting interface performance specifications for digital VTC/VT and digital television. The VTC/VT

sub-working group of T1Q1.5 is developing a catalogue of video motion artifacts associated with video compression and the resultant effects on video quality. The motion artifacts that are most noticeable to the viewer and that show the most potential for being measured are reproduced in Table 1. The artifact, definition of the artifact, and examples of the artifact are listed in the table. Artifacts are most apparent when video motion is present. The information content of a video signal that contains moving and/or changing scenes may simply be too great for a fixed transmission data rate. In such cases, image pixel values may not be updated rapidly enough, resulting in noticeable artifacts. Additional video coding artifacts can be found in Murakami et al. (1988).

Probably the most noticeable and objectional motion artifact is resolution degradation. Normally, stationary objects are coded with relatively high spatial resolution. However, as soon as the object moves, blurring and/or jerky motion of the object is noticed. In cases of excessive motion such as during camera pans and zooms, very objectionable blocking artifacts may appear. Other image coding artifacts seen upon close inspection include edge busyness and image persistence.

Table 1. Common Video Compression Artifacts

Motion Artifact	Definition
1. Resolution Degradation	The deterioration of motion video such that the received video imagery has suffered a loss of spatio-temporal resolution.
Examples:	
Blocking	The received video imagery possesses rectangular or checkerboard patterns not present in the original.
Blurring/smearing	The received video imagery has lost edges and detail present in the original.
Jerkiness	The original smooth and continuous motion is perceived as a series of distinct snapshots.
2. Edge Busyness	The deterioration of motion video such that the outlines of moving objects are displayed with randomly varying activity.
Example:	
Mosquito noise	The quantizing noise generated by the block processing of moving objects that gives the appearance of false small moving objects (e.g. a mosquito flying around a person's head and shoulders).
3. Image persistence	The appearance of earlier faded video frames of a moving and/or changing object within the current video frame.
Example:	
Erasure	An object that was erased continues to appear in the received video imagery.

2.2 Desirable Properties of Features

For the video quality measurement system shown in Figure 1, developing a set of sensitive and relevant features can be very difficult. Often, intuition and ad hoc procedures must be used to obtain a set of features which are meaningful and easily computed. The following list details some desirable properties of objectively measured features. These properties were used to steer the development of a set of features for measuring the quality of digitally transmitted video.

1. Correlation with subjective quality

   Perhaps the most critical attribute of a meaningful feature is strong correlation of the measured feature value with the subjective rating. If overall subjective ratings are not available, features should at least be sensitive to the amount of subjectively noticed video artifacts. The feature value should change monotonically when the amount of the artifact or distortion is increased.

2. Automation

   Feature extraction should be performable by an autonomous measurement system. Advantages include automatic detection of transmission line impairments, cost effective monitoring, and repeatability.

3. Application to many types of scenes

   Since the performance of the digital compression and transmission algorithm normally depends upon the type of imagery which is being compressed, the feature extraction procedure should be applicable to arbitrary video scenes. Thus, to test the video quality performance for a specific user application, one must use the appropriate type of video scenes.

4. Application as a local estimate

   There is evidence that the human viewer may determine the quality of a video scene by rating the quality of local details within the video scene (Westernik and Roufs, 1988).

Thus, the human viewer will often look at high contrast edges and contours to perform quality judgments. To account for this phenomena, feature extraction methods should take into account local or sub-regional properties (in space and/or time) of the video. Local estimates of quality may also be utilized by video compression algorithms to allocate bits dynamically to each sub-region of the video image.

5. Computational efficiency

   Features that are rapidly computed from the image are preferable from a cost and implementation standpoint. At best, the feature should be computable in real time, given reasonable hardware. Computationally efficient features may also be required for large, higher resolution imagery, such as HDTV.

6. Stability

   The feature should not be sensitive to distortions which the human viewer does not notice. For example, the feature should not be sensitive to small shifts in the mean of the video imagery nor other image distortions which fall below the threshold of visibility.

7. Functional independence

   When choosing a feature set, every feature within the set should convey different information. If a particular feature can be obtained as a function of other features within the feature set, that feature does not convey any additional information and can be disregarded.

8. Technology independence

   The feature is useful for a wide range of technologies. For instance, a feature developed for measuring digital image compression artifacts should also be useful in measuring video quality of an analog transmission channel.

2.3 Alignment Of Original And Distorted Video Imagery

Video imagery consists of a series of frames that are transmitted and displayed in sequence on a video display device. The most common video format in use in the United States is the National Television Systems Committee (NTSC) broadcast standard. With NTSC format, one frame consists of two sequential interlaced fields (Fink, 1975). The field scanning sequence is horizontally left to right, and vertically top to bottom. The first field scans the even numbered lines (2, 4, 6, etc.) and then the second field scans the odd numbered lines (1, 3, 5, etc.). To be able to time align input and distorted output video, the video digitizing system must capture each NTSC field (which occur at the rate of 59.94 fields per second). Some feature extraction techniques require that the input and distorted output video have been aligned beforehand.

Alignment or matching of input and distorted output video frames is complicated by the wide range of video coding schemes that are in use, and by the presence of an unknown video delay within the system under test. One common video compression scheme omits fields and/or frames before transmission, and then uses field and/or frame repetition on the receiving end to fill in the missing fields and/or frames. Thus, one is not guaranteed that an aligned output frame exists for each input frame. Sections 2.3.1 and 2.3.2 describe two methods for automatically aligning video scenes. Each method has been found to be useful, depending upon which features one desires to extract from the digitized video. Both alignment methods assume that some motion or changing scenery is present in the video. For completely static video scenes, alignment is not an issue.

2.3.1 Single-frame Temporal Alignment

Alignment of input and distorted output video scenes based on one output video frame is computationally fast and particularly useful when one wishes to preserve the temporal nature of the video. As was previously mentioned, because of the possibility of frame omission and repetition, there is no guarantee that an aligned output video frame exists for each input video frame. Therefore, it is necessary to align the input to the output, and not visa-versa. In other words, given an output frame, find the input frame which best matches that output frame. For single-frame temporal alignment, the alignment is only performed for

one output frame in the video sequence. The rest of the input and output video frames are temporally paired one for one, based upon the alignment found for the chosen output video frame. In practice, to assure that a causal alignment between the output and input video is obtained, the alignment for each of several consecutive output frames should be found. Then, the output frame which yields the smallest positive shift in time of the input video sequence produces the correct causal alignment.

The best matching input frame (for the chosen output frame) is found by computing the error difference images between the selected output frame and all reasonable input frames. When selecting the set of reasonable input frames, one must account for video delay within the system and the uncertainty of that video delay. Assuming the video scene contains some motion, the standard deviation of the error (accumulated over all pixels in the error image) goes to a minimum for the best aligned input image. The reader is referred to equation 1 of Appendix A for a mathematical definition of single-frame temporal alignment. The mean of the error image, being sensitive to small low frequency spectral components near DC, should not be used to perform time alignment. The standard deviation is not sensitive to small changes in the average gray level of the sampled images, but may be sensitive to changes in video gain. Thus, for this alignment technique (as well as for other feature extraction techniques proposed in this report), the gain of the video system should be stable over time.

A priori knowledge of the video delay for the system under test can ease the computational burden of the alignment process by minimizing the number of error difference images that must be examined. For each error difference image, computation of the standard deviation requires the accumulation of the image pixel values and the squares of the image pixel values. A computationally faster alignment could be obtained if the standard deviation calculation were replaced with a pixel counting scheme where one simply counted the number of error image pixel values that were less than a lower threshold or greater than an upper threshold. Here, care must be taken to make sure that any shifts in the mean of the error image are contained between the lower and upper thresholds.

Single-frame temporal alignment can be assisted if one is able to superimpose a time code or other timing data onto the input video frames. Then, alignment can be determined by processing a much smaller portion of the video image (just the part which contains the time code).

However, with this technique some accuracy may be lost since the video device under test might behave differently for the sub-regional part of the image that contains the changing time code.

In summary, single-frame temporal alignment preserves the temporal characteristics of the input and output video. The two contiguous sequences of input and output video frames are time aligned. All input and output video frames are preserved in the aligned sequences. Later in this report, single-frame alignment will be required before extracting temporal features of motion video like jerkiness (see Table 1).

2.3.2 Multi-frame Temporal Alignment

There are cases when the single-frame alignment technique is not adequate to perform the desired feature extraction. Such a case occurs when the user desires to measure the "snapshot" quality of the video imagery. For example, the user may require very high spatial resolution of the presented picture to troubleshoot circuit diagrams, but frequent updating of the video image may not be required. For a fixed transmission bit rate, the user may prefer one new high resolution video frame per second rather than thirty low resolution video frames per second. Another alignment technique, called multi-frame temporal alignment, is useful for features designed to measure the "snapshot" quality of the video system.

Multi-frame alignment differs from single-frame alignment in that the best matching input frame is found for <u>every</u> output frame. The techniques discussed for single-frame alignment are simply applied to each output frame. Since frames may have been omitted in the output video, multi-frame alignment will skip the video input frames that have no corresponding output frames. The computational task of multi-frame alignment may be eased considerably by intelligently choosing the set of input frames that must be examined for each output frame. In particular, the correct input frame alignment found for the previous output frames can be used to guess the input frame alignment for the current output frame.

A side benefit of multi-frame alignment is the detection of missing fields and/or frames in the output video. Multi-frame alignment may be used to compute the missing frame ratio (MFR), a useful measure of motion jerkiness. The MFR feature is computed as the number of missing frames

in the output video scene divided by the total number of frames (see equation 2 of Appendix A for a mathematical definition of MFR).

Figures 2 and 3 illustrate single-frame and multi-frame alignment applied to a video scene that contained motion. The top row of Figure 2 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 2. The solid lines in Figure 2 show the ordering of the input and output video frames when single-frame alignment was applied using the first codec output frame. The dashed lines show the ordering of the input and output video frames when multi-frame alignment was used. Figure 3 shows the error difference images (input frame minus output frame) that were used to determine the single-frame and multi-frame alignment of Figure 2. In Figure 3, white and black are positive and negative error, respectively, while the gray background represents no error. The top row in Figure 3 shows the error difference images between the four input frames (top row of Figure 2) and the first codec output frame (bottom, left image in Figure 2). Of the four error images in the top row of Figure 3, the first one (leftmost) contains the smallest error (least amount of black and white). Thus, when single-frame alignment was applied using the first codec output frame, the solid lines in Figure 2 give the pairing of the input and output video frames. Rows two, three, and four of Figure 3 give the corresponding error difference images for the second, third, and fourth codec output frames in Figure 2. 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. The missing frame ratio (MFR) for the example in Figures 2 and 3 is calculated as two divided by four (or .5), since two of the four input video frames were missing in the output.

In summary, multi-frame temporal alignment may destroy the original ordering of the input video sequence. Since the closest matching input video frame is found for each output video frame, some input video frames may be discarded. Multi-frame alignment is useful for developing quality measures that are independent of the output video frame rate. Such measures are useful for application groups that require high quality "snapshot" video at low frame rates (for instance, medical imaging). Later in this report, multi-frame alignment will be required before

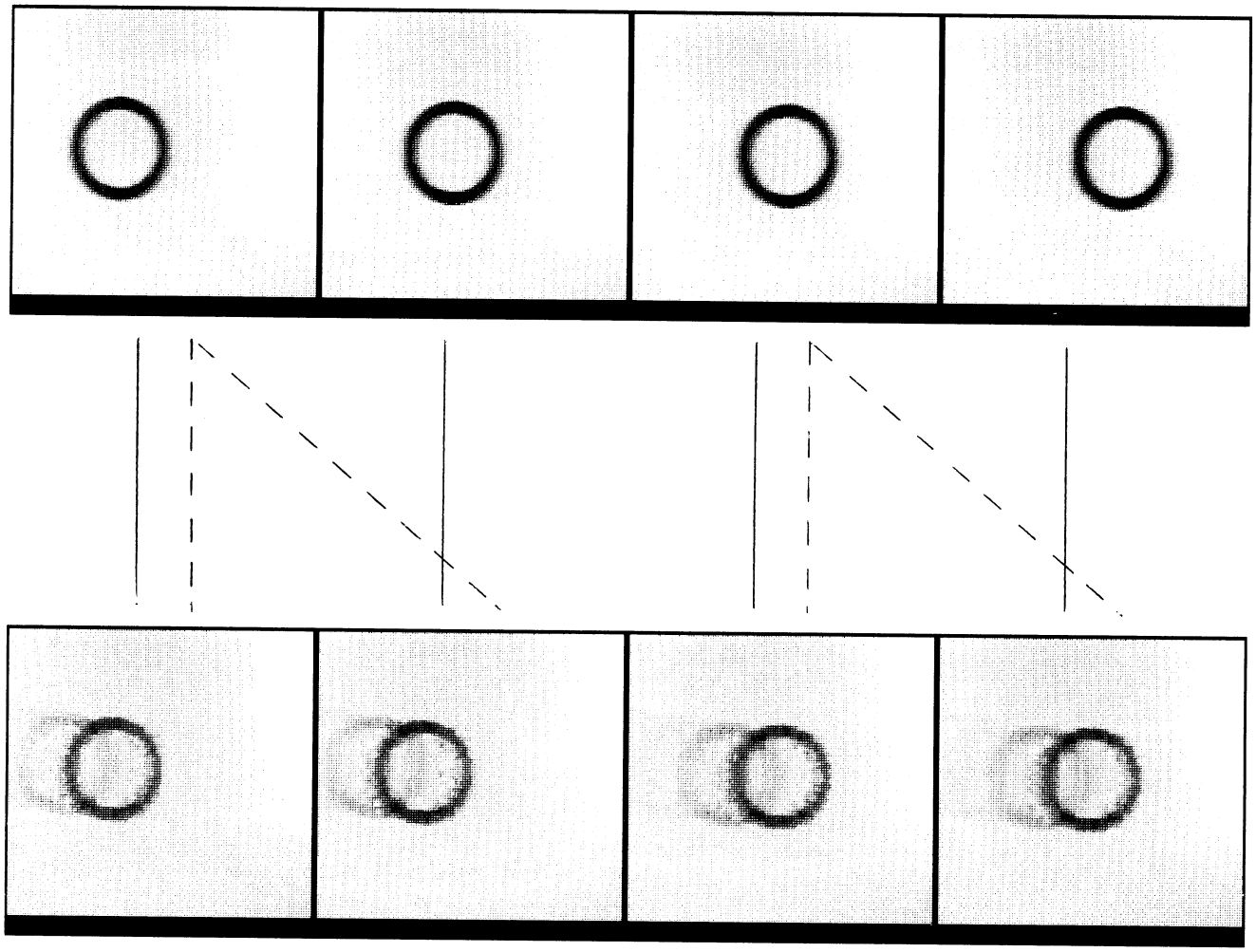

Figure 2. Single-frame and multi-frame alignment. Top row - original NTSC grabbed every 1/30 sec from left to right. Bottom row - VTC/VT codec output. Solid lines represent single-frame alignment and dashed lines represent multi-frame alignment.

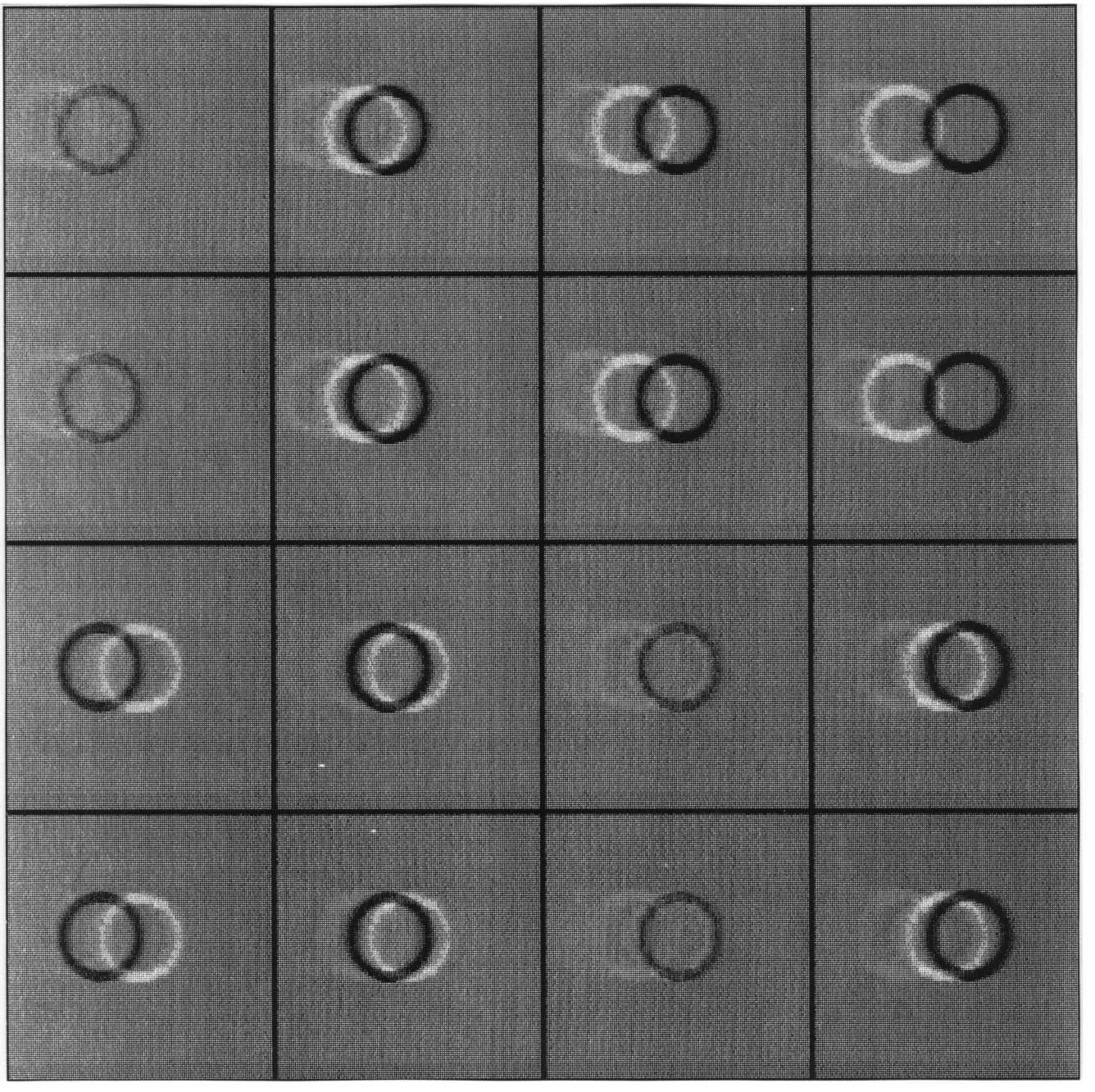

Figure 3. Error difference images (input–output) of Figure 2. Top row – NTSC input (top row in Figure 2) minus codec output image 1 (bottom row, leftmost frame in Figure 2). Second, third, and fourth rows are NTSC input minus codec output images 2, 3, and 4, respectively.

extracting spatial blurring, blocking, and edge busyness (see Table 1) features that accurately measure the "snapshot" video quality.

2.4 Preconditioning Of The Sampled Video

Certain spatial-temporal properties of the video display and/or human visual system may be taken into account by proper preconditioning of the sampled video before feature extraction. Image preconditioning normally involves application of some form of non-linear amplitude and/or frequency domain weighting functions. Historically, the goal of image preconditioning has been to enable distortion measures (such as the error difference) to correlate accurately with the subjective quality rating. Mannos and Sakrison (1974), Sakrison (1977), Limb (1979), Carlson and Cohen (1980), Barten (1987, 1988), Miyahara (1988), and Ohtsuka et al. (1988) have suggested possible amplitude and frequency domain weighting functions for black and white pictures and/or video displays. Amplitude domain transformations have also been suggested for color images. The red, green, and blue color system typically employed in video displays does not yield a perceptually uniform color space. Ideally, in a perceptually uniform color space, each color axis is perceptually independent of the others and psychometrically uniform. The Munsell color space (Newhall, 1943), the CIE color space (CIE Supplement No. 2 to CIE Publication No. 15, 1978), and transformations proposed by Miyahara and Yoshida (1988), and Taylor et al. (1989) are such uniform color spaces. Frequency domain transformations for color images have not been addressed and are currently a research topic.

A subjectively judged video library that contains the wide range of impairments found in digitally transmitted video systems is required to evaluate the usefulness of the various weighting functions. Implementation of amplitude domain weighting functions is normally computationally efficient. Implementation of frequency domain weighting functions is computationally expensive as two fast Fourier transforms (FFT) per image are required (one forward and one inverse). For this report, no preconditioning (other than that described for the extraction of each individual feature) has been performed.

2.5 Spatial Blurring Features

Spatial resolution degradation is an artifact that normally occurs when a video camera is imperfectly focused or when motion is present in a video scene. Camera defocusing reduces spatial resolution by spreading incident light over a larger surface area. Thus, a defocused camera is unable to pass the high spatial frequency information present in imagery containing sharply defined edges and fine detail. Under conditions of video motion, the bandwidth compression techniques typically employed in digital video systems are unable to retain enough of the high frequency information to avoid blurring of the edges. Investigators in the fields of human vision and human object recognition have recognized the importance of sharp edges for correct visual perception and recognition of objects (Shapley and Tolhurst, 1973, Held et al., 1978, Geuen and Preuth, 1982, Beiderman, 1985, Owens et al., 1989). The importance of sharp edges for moving objects is currently a research topic and appears to depend on whether or not the eye can track the object. Several methods have been proposed to detect automatically the sharpness of image edges. In section 2.5.1, the procedure of Toit and Lourens (1988) for estimating the edge sharpness of arbitrary video imagery has been adapted to measuring the spatial resolution degradation present in digitally transmitted video systems.

2.5.1 Feature Extraction Technique

A method for estimating the sharpness of edges in sampled video imagery can be obtained using very simple image processing techniques. The procedure relies on being able to sample the input (undistorted) video imagery as well as the output (distorted) video imagery. The input video imagery is required as a reference so that the amount of spatial resolution degradation present in the output imagery can be estimated. The edge sharpness feature can be extracted by computing the amount of energy present in the edge extracted video imagery. The theory is that sharper edges will contribute more high intensity pixel values than blurred edges. Several steps are required to apply the technique:

1. Video alignment

   If one desires to observe the instantaneous value of the feature (frame-by-frame), then single-frame temporal alignment of the input and output video is recommended. Strictly

speaking, time alignment of the input and output video is not required to extract this feature, provided one only requires the average value of the feature (over all frames in the video sequence).

2. Video preconditioning

The sampled video imagery is preconditioned to remove edge energy contributions resulting from camera interlace effects and noise spikes. Because edge extraction filters (to be applied in step 3 below) involve taking the difference of neighboring image pixel values, they will enhance noise in the imagery as well as edges. Therefore, some preconditioning of the imagery is strongly desired before application of the edge extraction filter.

If noise spikes are present, they can be removed by use of a small median filter that does not significantly blur the edges (Tzafestas, 1986, Gonzalez and Wintz, 1987, Jain, 1989). Fine detail, such as object corners, will be blurred by the median filter. For the imagery to be presented later in this report, median filtering was performed (see Appendix B for a description of the median filter that was used).

Camera interlace effects may be present in an NTSC video system when the video scene contains objects that are moving horizontally. In Figure 4, vertical edges are seen to form many alternate horizontal edges that have a length proportional to the velocity of horizontal motion. The horizontal edges caused by camera interlace will contribute a large amount of erroneous edge energy. Here, the interlace effects could be removed by sub-sampling the image by a factor of 2 in both horizontal and vertical directions (every other row and column in the image being discarded). A more desirable method of reducing the erroneous edge energy due to interlace would be to select an edge extraction filter which is insensitive to interlace. This is the recommended method and the one which is used here in step 3 below. The Sobel edge extraction filter (Tzafestas,

1986, Gonzalez and Wintz, 1987, Jain, 1989) is insensitive to interlace effects because it detects edges by computing the difference between image pixel values that are spaced two pixels apart (see Appendix B for a description of the Sobel filter). Thus, the Sobel filter does not extract the edges due to alternating black and white interlace lines (as shown in Figure 4).

(a) Slow Motion (b) Fast Motion

Figure 4. Camera interlace effects caused by horizontal motion.

3. Edge extraction

An edge extraction filter is applied to the preconditioned video imagery. The reader is referred to Gonzalez and Wintz (1987) or Jain (1989) for a description of several of the many different types of edge extraction filters. For the imagery to be presented later in this report, a Sobel filter was chosen.

4. Feature computation

Several features can be computed from the edge extracted or Sobel filtered imagery. Four are suggested here:

a. The mean of the Sobel image (M-SI)

M-SI is computed as the summation of the image pixel values divided by the total number of pixels. Here, the summation can be performed over any sub-regional area of the image. See Appendix A, equation 3 for a mathematical definition of M-SI.

b. The standard deviation of the Sobel image (SD-SI)

SD-SI is computed as the square root of (the summation of the squares of the image pixel values divided by the total number of pixels, minus the square of M-SI). This estimate of the standard deviation is asymptotically unbiased for a large number of image pixels, which is typically the case. See Appendix A, equation 4 for a mathematical definition of SD-SI.

c. The root mean square of the Sobel image (RMS-SI)

RMS-SI is computed as the square root of (the summation of the squares of the image pixel values divided by the total number of pixels). See Appendix A, equation 5 for a mathematical definition of RMS-SI.

d. The number of pixels greater than a threshold of the Sobel image (NPGT-SI)

NPGT-SI is computed as the total number of pixels within any sub-regional area that exceed a fixed threshold. Advantages of this feature include the ability to detect the blurring of just the sharpest edges, and ease of computation. There is some indication that humans may perform quality assessment by examining the sharpest high contrast edges (Westernik and Roufs, 1988). The higher the threshold for NPGT-SI, the sharper the edges must be before being included in the summation. Subjectively judged video data could be used to

determine the threshold setting that gives the best correspondence with subjective quality. Since subjective data was unavailable at the time of this report, a somewhat arbitrary threshold was selected that included the predominate edges of the image. See Appendix A, equation 6 for a mathematical definition of NPGT-SI.

The decrease in the amount of edge energy that the output imagery has with respect to the input imagery can be used to estimate the amount of spatial resolution degradation present in the output imagery with respect to the input imagery. Alternately, since the input imagery contains sharper edges (and hence higher pixel values) than the output imagery, the decrease in the number of pixel values that exceed the threshold can be used to estimate the spatial resolution degradation. In either case, by normalizing with the reference imagery, a feature can be formed that varies between 1 (no edge blurring) to 0 (complete edge blurring).

The features described above exhibit many of the desirable properties of features mentioned earlier. In particular, the features are applicable to arbitrary video scenes and may be applied to any sub-region of the image, making them useful for local estimates of image quality. In addition, the feature extraction process is computationally efficient and stable. The features are insensitive to noise spikes (due to median filtering) and small gray level shifts in the image (since edges are computed from the differences of neighboring image pixel values and thus, the background gray level subtracts off).

2.5.2 Sample VTC/VT Results

For illustrative purposes, the method for extracting the spatial blurring features was applied to several VTC/VT video frames sampled by an 8 bit video frame grabber. Figure 5 shows the sampled imagery after median filtering. The top left image was captured from an NTSC camera with no motion present in the video scene. The top right image was captured from an NTSC camera when rotational motion was present in the video scene. The bottom right image was captured from the output of a VTC/VT codec running at the digital signal 1 (DS1) rate of 1.544 Mbps, where the input imagery to the codec was the same as that shown in the

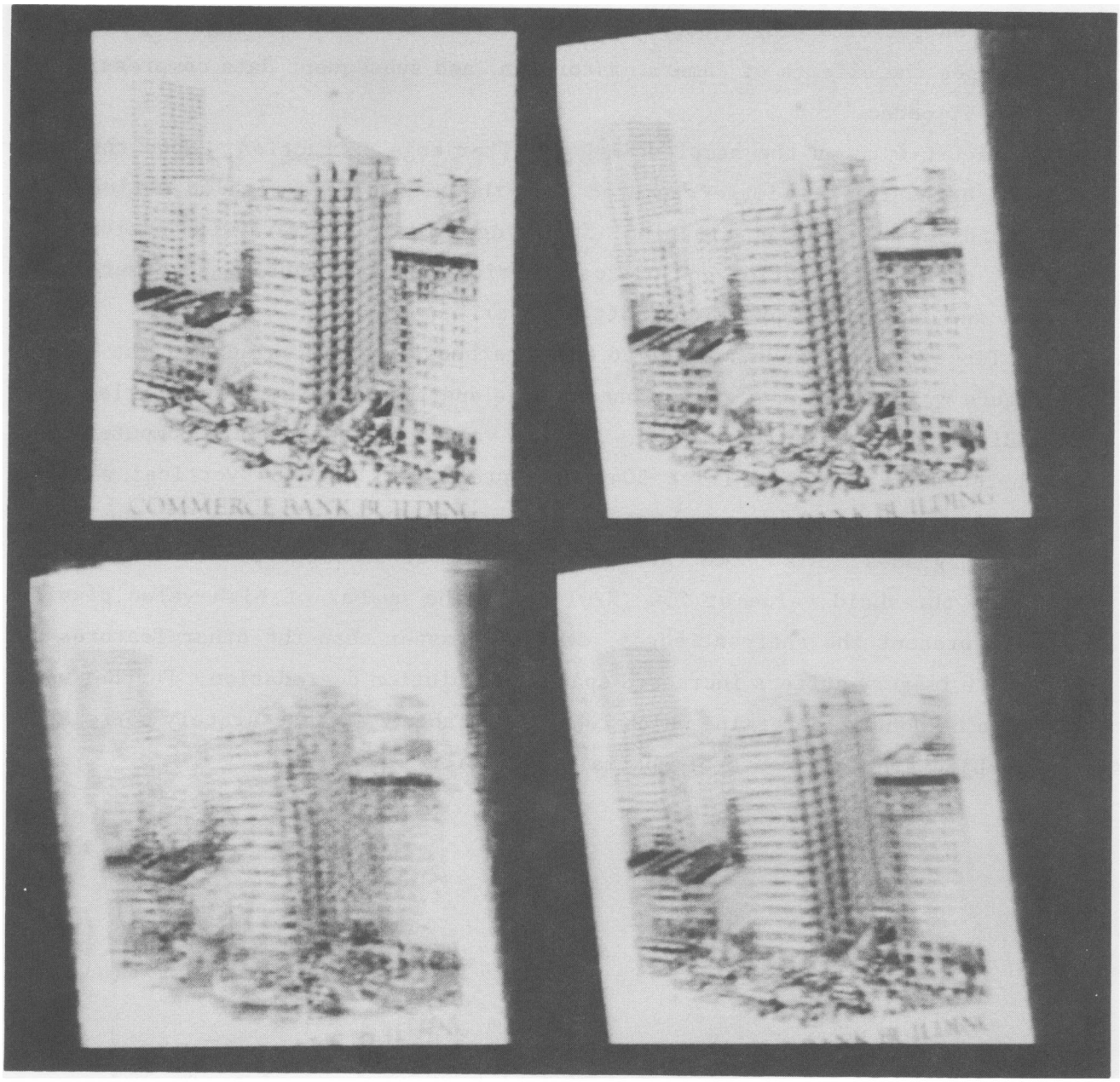

Figure 5. VTC/VT imagery containing rotational motion. Top left – camera with no motion. Top right – camera with rotational motion. Bottom right – camera with motion and DS1 data compression. Bottom left – camera with motion and 1/4 DS1 data compression.

top right image of Figure 5. The bottom left image was captured from the output of a VTC/VT codec running at rate 1/4 DS1. In Figure 5, one can clearly see the effects of camera distortion, and subsequent data compression by the VTC/VT codec.

Figure 6 shows the sampled imagery after edge extraction. Note the well defined edges for the image captured from the NTSC camera with no motion (top left) and the successive worsening of the edge blur for camera with motion (top right), camera with motion and DS1 compression (bottom right), and camera with motion and 1/4 DS1 compression (bottom left).

Table 2 shows the unnormalized edge sharpness feature values for the images in Figure 6. To eliminate the erroneous edge energy at the image boundaries (due to median and Sobel filtering), the feature values in Table 2 were computed over a sub-rectangular region (size 504 horizontal pixels by 464 vertical pixels) centered on the main image. Note the decrease in edge energy as the image quality degrades. Also note the decreasing number of image pixels that exceed a chosen threshold value of 250 (NPGT-SI). The number of high value pixels, which represent the sharpest edges, decrease faster than the other features as the input imagery suffers increased spatial resolution degradation. Further work needs to be done to determine which feature in Table 2 most accurately correlates with subjective judgements of spatial resolution.

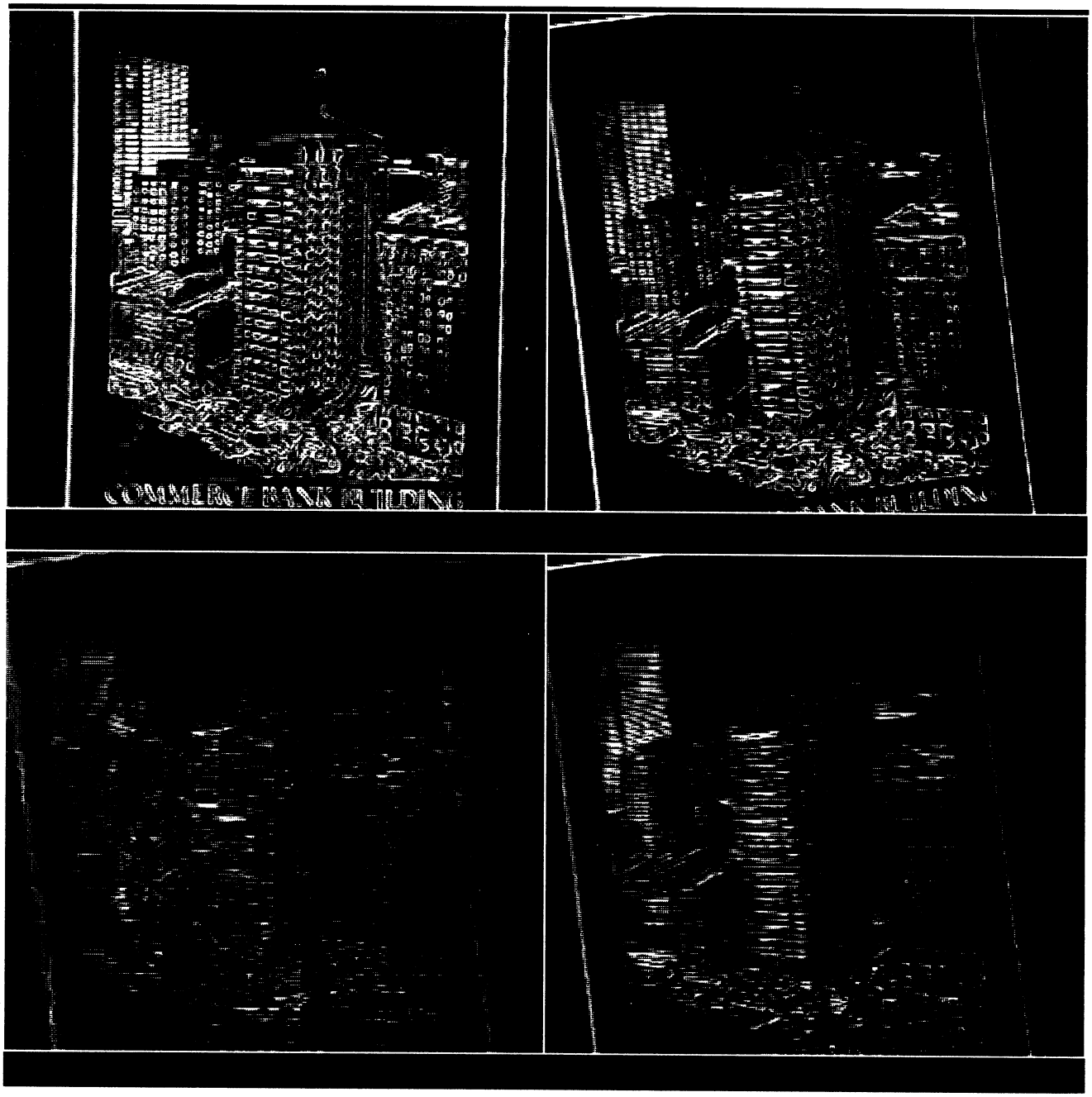

Figure 6. Sobel filtered edge extracted VTC/VT imagery of Figure 5. Note the well defined edges for the image captured from an NTSC camera with no motion (top left) and the successive worsening of the edge blur for camera with motion (top right), camera with motion and DS1 compression (bottom right), and camera with motion and 1/4 DS1 compression (bottom left).

Table 2. Spatial Blurring Features For VTC/VT Imagery Of Figure 6

Image	M-SI	SD-SI	RMS-SI	NPGT-SI
Top Left (Still)	59.6	81.2	100.8	9116
Top Right (Camera + rotation)	48.2	64.7	80.7	3882
Bottom Right (Camera + rotation + DS1)	37.3	48.6	61.3	995
Bottom Left (Camera + rotation + 1/4 DS1)	32.2	36.3	48.6	184

The edge sharpness features were also computed for a typical VTC/VT scene that contained upper body motion. The video scene in Figure 7 contains motion of the man's head and hands, and the report that the man is holding. The top row shows a sample of 4 consecutive images that were frame-grabbed from the original NTSC VTC/VT scene. The images were grabbed at each field increment of the video recorder. Thus, the time difference between consecutive images in a row is approximately 1/60 of a second. The second, third, and fourth rows of images were obtained from the output of a VTC/VT codec that compressed the NTSC video to bit rates of DS1, 1/2 DS1, and 1/4 DS1, respectively. The single-frame temporal alignment method has been used to align the codec output video shown in Figure 7. For clarity, the images in the first column (leftmost) of Figure 7 have been expanded in Figure 8. In Figure 8, the top left image is the original NTSC, the top right is DS1, the bottom right is 1/2 DS1, and the bottom left is 1/4 DS1. Note that most of the image distortion occurs locally in areas that contain motion (man's right hand), and that the static background is relatively distortion free. As the codec is forced to operate at lower bit rates, areas of the image that contain motion become more and more blurred.

Figure 7. VTC/VT imagery containing upper body motion. Top row - NTSC input. Second row -codec output at rate DS1. Third row - codec output at rate 1/2 DS1. Bottom row - codec output at rate 1/4 DS1.

Figure 8. Leftmost column of Figure 7 expanded. Top left - NTSC input. Top right - codec output at rate DS1. Bottom right - codec output at rate 1/2 DS1. Bottom left - codec output at rate 1/4 DS1.

Figure 9 shows the video of Figure 7 after median filtering and edge extraction. For clarity, Figure 10 shows the expanded video of Figure 8 after median filtering and edge extraction. Note that edges of moving objects appear less intense as the codec is forced to operate at lower bit rates. Thus the edges are most blurred for images in the bottom row (bit rate of 1/4 DS1). Table 3 shows the average of the unnormalized spatial blurring features for eight consecutive images, the first four of which are shown in Figure 9. The sub-rectangular image regions and thresholds (for NPGT-SI) that were used to generate Table 2 were also used to generate Table 3. The features in Table 2 were generated from video imagery that contained rotational motion which included a large part of the image. The features in Table 3 were generated from video imagery that contained only a small amount of natural motion. The codec performs differently for the two types of video scenes, and this is reflected in the computed features.

Since subjective quality ratings are based on a video scene that is normally 10 seconds long, a very robust process would be to extract the features from many frames of video, and even from many sub-regions of each video frame. Then, the feature classification system (shown in Figure 1) could utilize all of the feature samples to improve the video quality classification.

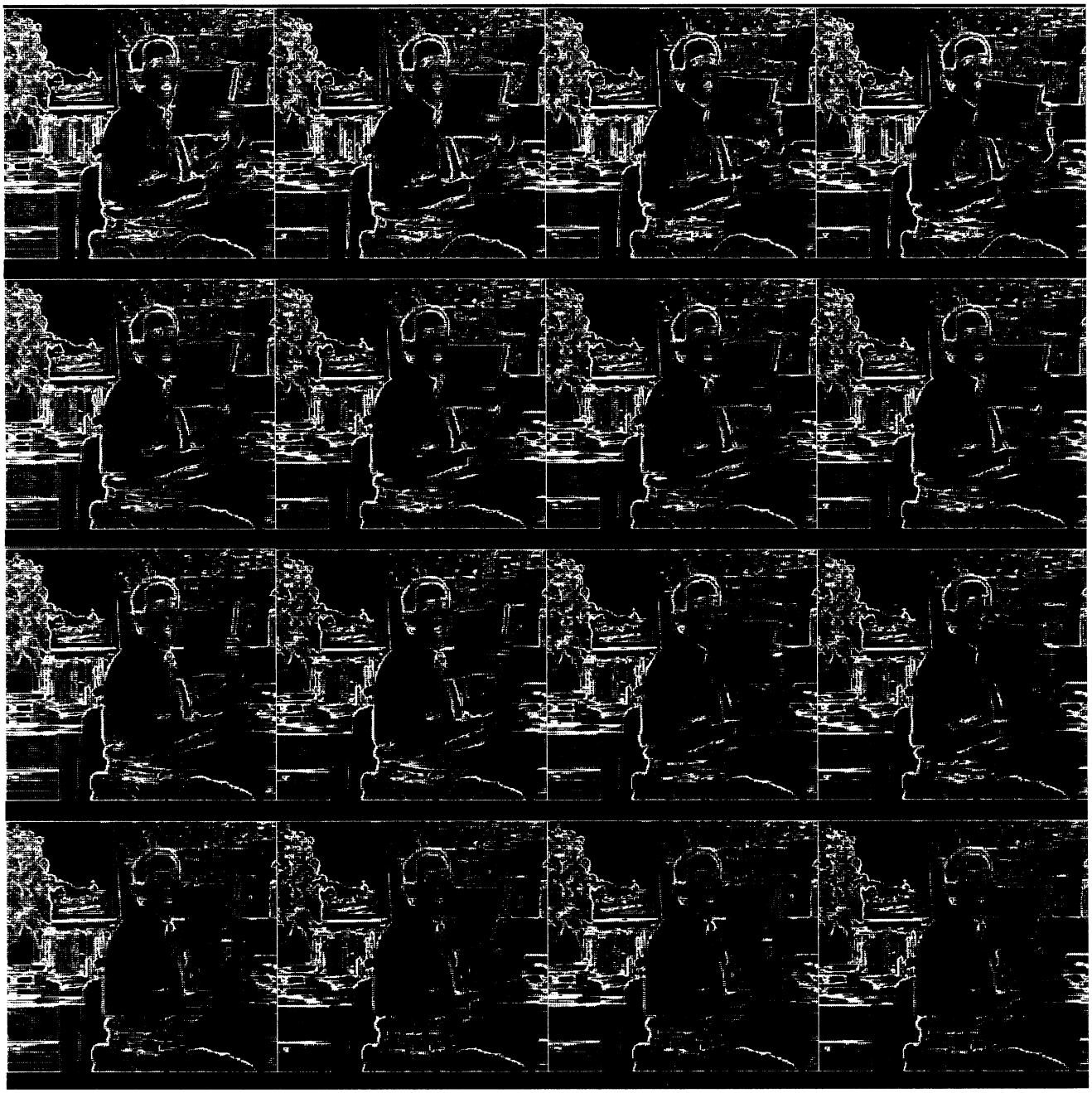

Figure 9. Sobel filtered edge extracted VTC/VT imagery of Figure 7. Note the well defined upper body edges for the images captured from the NTSC input (top row) and the successive worsening of the edge blur for DS1 compression (second row), 1/2 DS1 compression (third row), and 1/4 DS1 compression (bottom row).

Figure 10. Sobel filtered edge extracted VTC/VT imagery of Figure 8. Note the well defined upper body edges for the images captured from the NTSC input (top left) and the successive worsening of the edge blur for DS1 compression (top right), 1/2 DS1 compression (bottom right), and 1/4 DS1 compression (bottom left).

Table 3. Spatial Blurring Features For VTC/VT Imagery Of Figure 9

Scene	M-SI	SD-SI	RMS-SI	NPGT-SI
Top Row (NTSC)	70.4	103.3	125.1	14388
Second Row (DS1)	61.7	90.0	109.1	10247
Third Row (1/2 DS1)	60.9	89.5	108.3	9972
Bottom Row (1/4 DS1)	59.2	83.8	102.6	8265

2.6 Blocking, Edge Busyness, and Image Persistence Features

Blocking, defined in Table 1, is a severe form of spatial resolution degradation that normally occurs at low codec bit rates when there is a lot of motion in some sub-region or all of the video scene (such as during camera pans or zooms). Edge busyness and image persistence, also defined in Table 1, are video coding artifacts that causes false activity to appear around edges or elsewhere is the video scene. Blocking, edge busyness, and image persistence are most noticeable when the motion involves a high contrast (sharp) edge. Blocking, edge busyness, and image persistence cause edge energy to appear in the output video scene that was not present in the original input video scene. Human viewers have semantic knowledge of how certain items should look and they take objection to the presence of erroneous, out of place artifacts such as blocking, edge busyness, and image persistence. In particular, the appearance of false regular edge energy such as blocking is very noticeable and objectionable to the human viewer (more so than spatial blurring). Therefore, it is desirable to have a set of features that only measures the amount of false edge energy in a video scene.

Section 2.6.1 proposes a technique for extracting a set of features that quantitatively measures the amount of false edge energy in the output video scene. The features may be used to measure blocking, edge busyness, and image persistence since all contribute false edge energy to the output video scene. A by-product of the false edge energy feature is another set of features for measuring spatial blurring. The new

measures for spatial blurring, to be described below, do not contain false edge energy. Thus, the new measures for spatial blurring are more accurate than the measures presented in section 2.5.1 (M-SI, SD-SI, RMS-SI, NPGT-SI) when there is a large amount of false edge energy present in the video scene. The ability to separately measure spatial blurring and false edge energy may be important since each has a unique affect on perceived video quality. A video quality assessment system that detects the presence of blocking, a very objectionable artifact, could heavily penalize the overall quality rating.

2.6.1 Feature Extraction Technique

Edges in output video that are less intense than the corresponding edges in the input video are considered to have been blurred. False edge energy, due to the presence of such artifacts as blocking, edge busyness, or image persistence appears in the distorted output video but not in the undistorted input video. Suppose one were to compute an edge error image by subtracting the edge filtered output image from the corresponding edge filtered input image. Then, positive pixel values would be obtained for blurred output edges since, by definition, the edges in the input image are more intense (higher value) than the corresponding edges in the output image. Likewise, negative pixel values would be obtained for false output edges, since false edges in the output image are more intense than the corresponding edges in the input image. Thus, one could form an edge error image in which positive error represents blurring and negative error represents false edges. The exact feature extraction technique is given below.

1. Video alignment

   Multi-frame temporal alignment of the input and output video is required since the edges in the input and output images must be properly aligned. Thus, the extracted features will be representative of the "snapshot" performance of the video system under test since one is always comparing the output image with the closest matching input image. The single-frame temporal alignment method is not used because feature errors due to alignment could be generated, particularly if fields and/or frames have been omitted in the output video.

2. Video preconditioning

The sampled video imagery is preconditioned as previously described in section 2.5.1.

3. Edge extraction

An edge extraction filter is applied to the preconditioned video imagery as previously described in section 2.5.1. Here, a Sobel edge extraction filter was used (described in Appendix B).

4. Difference image

For each output/input video frame pair of interest, the Sobel difference image is computed as the Sobel filtered input image minus the Sobel filtered output image.

5. Feature computation

Several features can be extracted from the Sobel difference image. Eight are suggested here. Four of the eight are blurring features since they are extracted from the positive pixel values of the Sobel difference image. The other four are false edge features since they are extracted from negative pixel values of the Sobel difference image. All eight features possess the same desirable properties of features as the spatial blurring features for single-frame temporal alignment.

a. The mean of the positive Sobel difference image (M-PSDI)

M-PSDI is computed as the summation of the positive image pixel values divided by the total number of pixels in the sub-regional area of the image. The total number of pixels in the sub-regional area is used as the divisor, rather than just the number of positive pixels, so that the total amount of blurring energy can be directly compared to the total amount of false edge

energy. See Appendix A, equation 7 for a mathematical definition of M-PSDI.

b. The standard deviation of the positive Sobel difference image (SD-PSDI)

SD-PSDI is computed as the square root of (the summation of the squares of the positive image pixel values divided by the total number of pixels, minus the square of M-PSDI). See Appendix A, equation 8 for a mathematical definition of SD-PSDI.

c. The root mean square of the positive Sobel difference image (RMS-PSDI)

RMS-PSDI is computed as the square root of (the summation of the squares of the positive image pixel values divided by the total number of pixels). See Appendix A, equation 9 for a mathematical definition of RMS-PSDI.

d. The number of pixels greater than a threshold of the positive Sobel difference image (NPGT-PSDI)

NPGT-PSDI is computed as the total number of pixels within any sub-regional area that exceed a fixed threshold. Advantages of this feature include the ability to measure the number of severely blurred pixels, and ease of computation. See Appendix A, equation 10 for a mathematical definition of NPGT-PSDI.

e. The mean of the negative Sobel difference image (M-NSDI)

M-NSDI is computed as the summation of the negative image pixel values divided by the total number of pixels in the sub-regional area of the image. See Appendix A, equation 11 for a mathematical definition of M-NSDI.

f. The standard deviation of the negative Sobel difference image (SD-NSDI)

SD-NSDI is computed as the square root of (the sum of the squares of the negative image pixel values divided by the total number of pixels, minus the square of M-NSDI). See Appendix A, equation 12 for a mathematical definition of SD-NSDI.

g. The root mean square of the negative Sobel difference image (RMS-NSDI)

RMS-NSDI is computed as the square root of (the sum of the squares of the negative image pixel values divided by the total number of pixels). See Appendix A, equation 13 for a mathematical definition of RMS-NSDI.

h. The number of pixels less than a threshold of the negative Sobel difference image (NPLT-NSDI)

NPLT-NSDI is computed as the total number of pixels within any sub-regional area that are less than a fixed threshold. Advantages of this feature include the ability to measure the number of pixels corrupted with severe false edges, and ease of computation. See Appendix A, equation 14 for a mathematical definition of NPLT-NSDI.

Normalization of the above eight features can be performed by dividing by the appropriate spatial blurring features of the undistorted input video (M-SI, SD-SI, RMS-SI, and NPGT-SI from section 2.5.1). Then the amount of blurring or false edges in the output video with respect to the input video is obtained. The thresholds for NPGT-PSDI and NPLT-NSDI determine the severity of the blurring or false edges that the user is interested in measuring and these thresholds do not have to be identical to each other nor to NPGT-SI. The choice of the three thresholds will determine the range of the normalized features.

2.6.2 Sample VTC/VT Results

For illustrative purposes, the four spatial blurring features and four false edge features (blocking, edge busyness, and image persistence) were extracted from sampled VTC/VT imagery of a moving black ring against a white background (motion was from left to right). The high contrast moving edges of the black ring provided a sufficiently complicated object for the VTC/VT codec under test to exhibit the blocking, edge busyness, and image persistence artifacts. The top right image in Figure 11 shows the median filtered output of the VTC/VT codec that was operating at a bit rate of 1/4 DS1. The top left image in Figure 11 is the corresponding original NTSC input image after median filtering (found by using multi-frame temporal alignment). Note that the codec output image exhibits blurring, blocking, edge busyness, and image persistence (see Table 1). The bottom left and right images of Figure 11 show the Sobel edge extracted imagery of the NTSC input and codec output, respectively.

Figure 12 shows the Sobel difference image found by subtracting the bottom right image of Figure 11 from the bottom left image of Figure 11. For display purposes only (not for feature value computations), Sobel difference image pixel values were linearly scaled such that pixel values of zero (no error) are shown as a gray shade of 128 in Figure 12. The gray shade of 128 fell halfway between black (0) and white (255) on the 8 bit video printer that was used to generate the image. Thus, pixels that appear white are due to blurred edges in the output and pixels that appear black are due to false edges in the output. Clearly, the blurring energy has been separated from the blocking, edge busyness, and image persistence energy.

The four unnormalized blurring features and four unnormalized false edge features for Figure 12 are shown in Tables 4 and 5, respectively. To eliminate the erroneous edge energy at the image boundaries (due to median and Sobel filtering), all feature values were computed over a sub-rectangular region (size 504 horizontal pixels by 464 vertical pixels) centered on the main image. For NPGT-PSDI, pixel values that exceeded a threshold of 125 were counted. For NPLT-NSDI, pixel values that were less than a negative threshold of -125 were counted. For comparison to the reference, the spatial blurring features for the original NTSC Sobel extracted image (Figure 11, bottom left) were M-SI=14.7, SD-SI=27.7, RMS-SI=31.3, NPGT-SI=558 (calculated using a threshold of 250). Note that there was more blurring energy than false edge energy for this example.

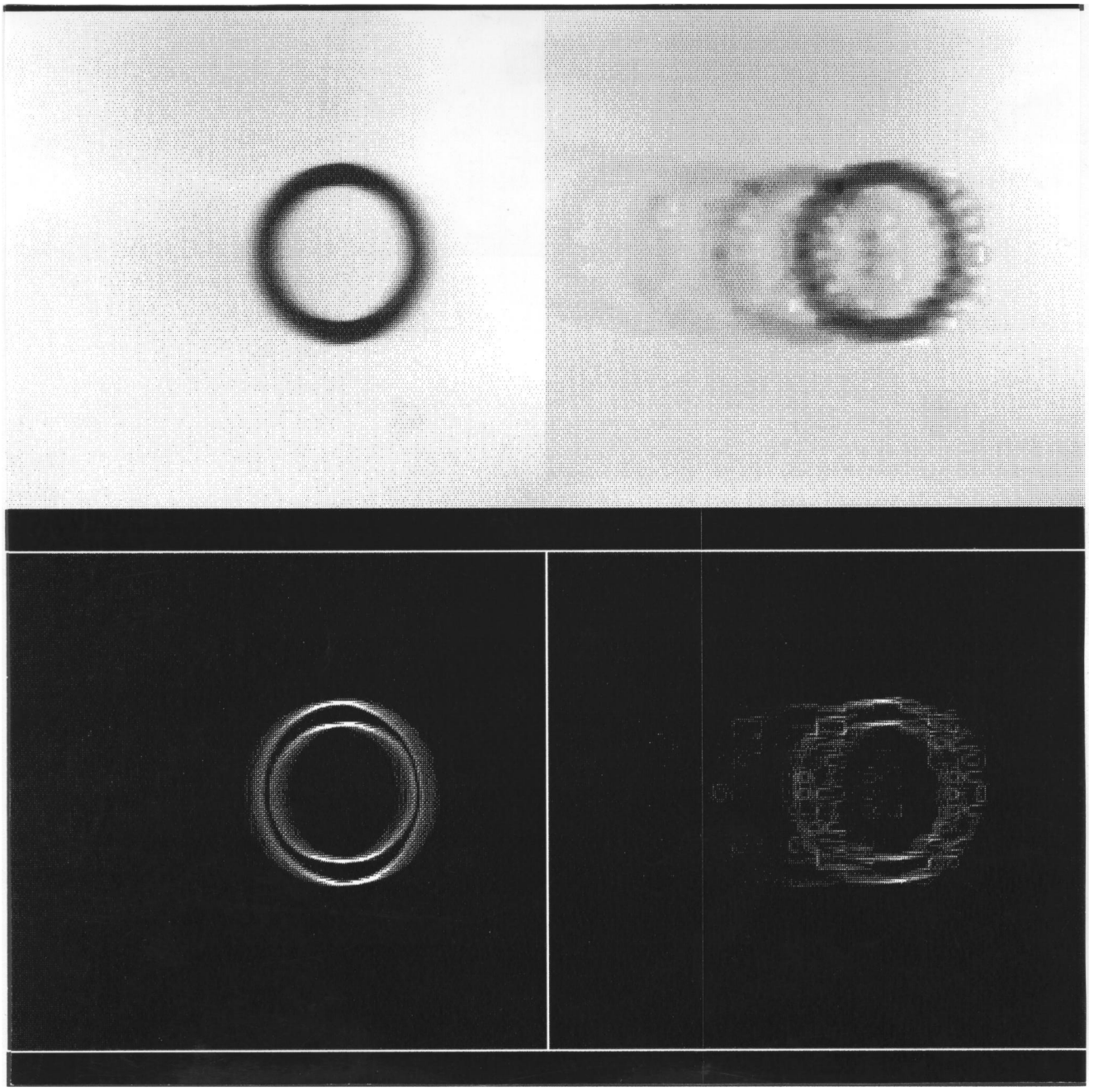

Figure 11. VTC/VT imagery of moving black ring against white background. Top left - median filtered NTSC input. Top right - median filtered codec output at rate 1/4 DS1. Bottom left - Sobel filtered NTSC. Bottom right - Sobel filtered codec output.

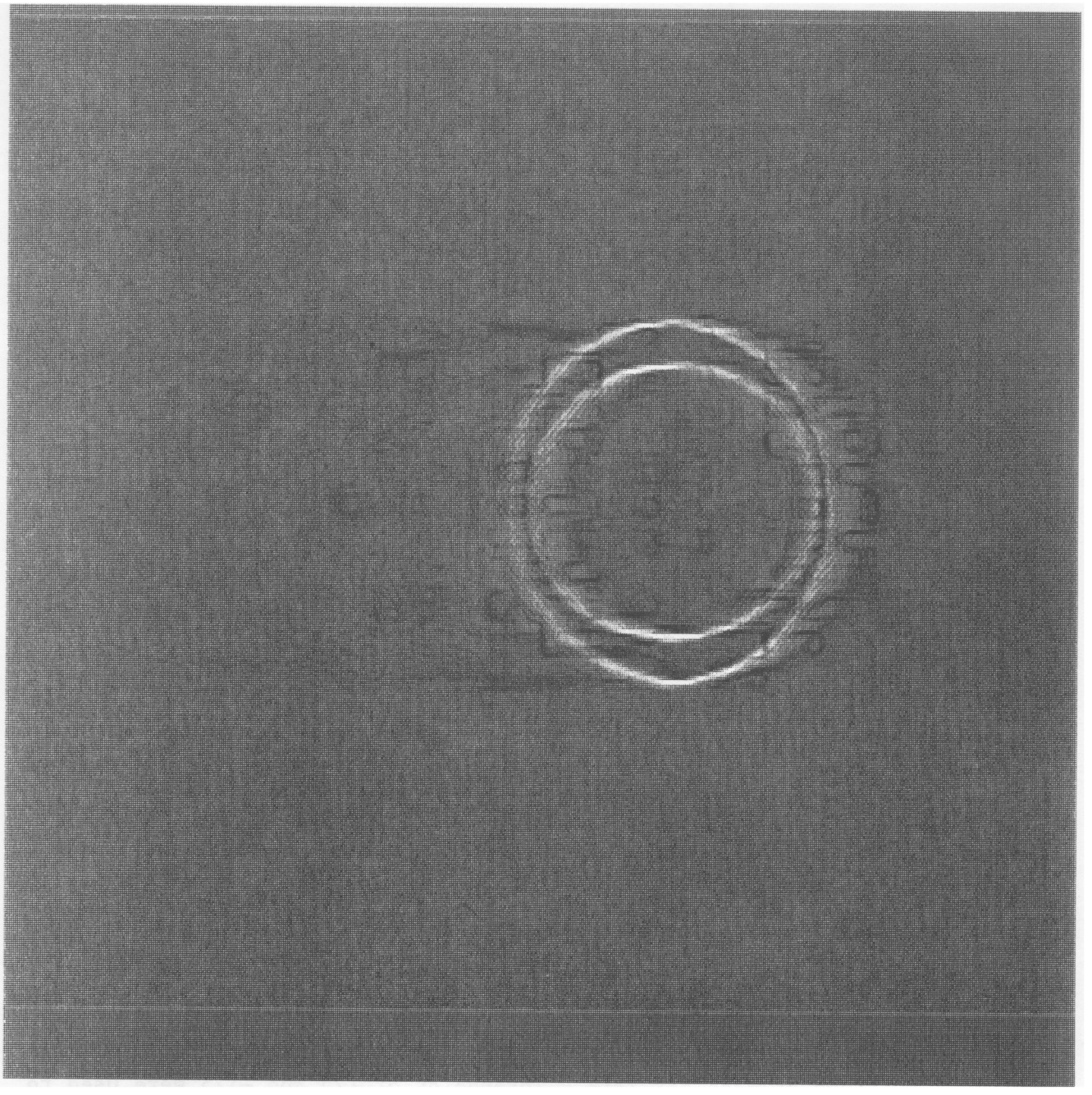

Figure 12. Sobel difference image of Figure 11. Note the separation of the blurring energy (white) from the blocking and edge busyness energy (black).

Table 4. Spatial Blurring Features For Figure 12

Scene	M-PSDI	SD-PSDI	RMS-PSDI	NPGT-PSDI
Moving Ring (1/4 DS1)	5.6	15.9	16.8	882

Table 5. False Edge Features For Figure 12

Scene	M-NSDI	SD-NSDI	RMS-NSDI	NPLT-NSDI
Moving Ring (1/4 DS1)	-5.1	9.3	10.6	26

Individual adjustment of the NPGT-PSDI and NPLT-NSDI thresholds can scale the importance of the blurring effects in relationship to the false edge effects.

Eight consecutive images of VTC/VT codec output video that contained upper body motion (the first four of which are shown in Figure 7) were processed to extract the spatial blurring, blocking, and edge busyness features. The corresponding NTSC input video frames for each of the codec output frames shown in rows 2, 3, and 4 of Figure 7 were found using multi-frame temporal alignment, rather than single-frame temporal alignment as shown in row 1 of Figure 7. Figure 13 shows the resulting Sobel difference images. The top, second, and bottom rows of Figure 13 were obtained by processing the codec output video for bit rates of DS1, 1/2 DS1, and 1/4 DS1, respectively. Tables 6 and 7 give the average of the unnormalized features, where the average was computed over 8 consecutive images at each bit rate. Thresholds of 125 and -125 were used to compute NPGT-PSDI in Table 6 and NPLT-NSDI in Table 7. The features in Tables 6 and 7 are directly comparable to the features in Table 3. Note from Figure 13 and Tables 6 and 7 that both blurring (white) and false edges (black) increased as the coding bit rate fell from DS1 to 1/2 DS1 to 1/4 DS1. Also note that there was more blurring than false edges at all bit rates.

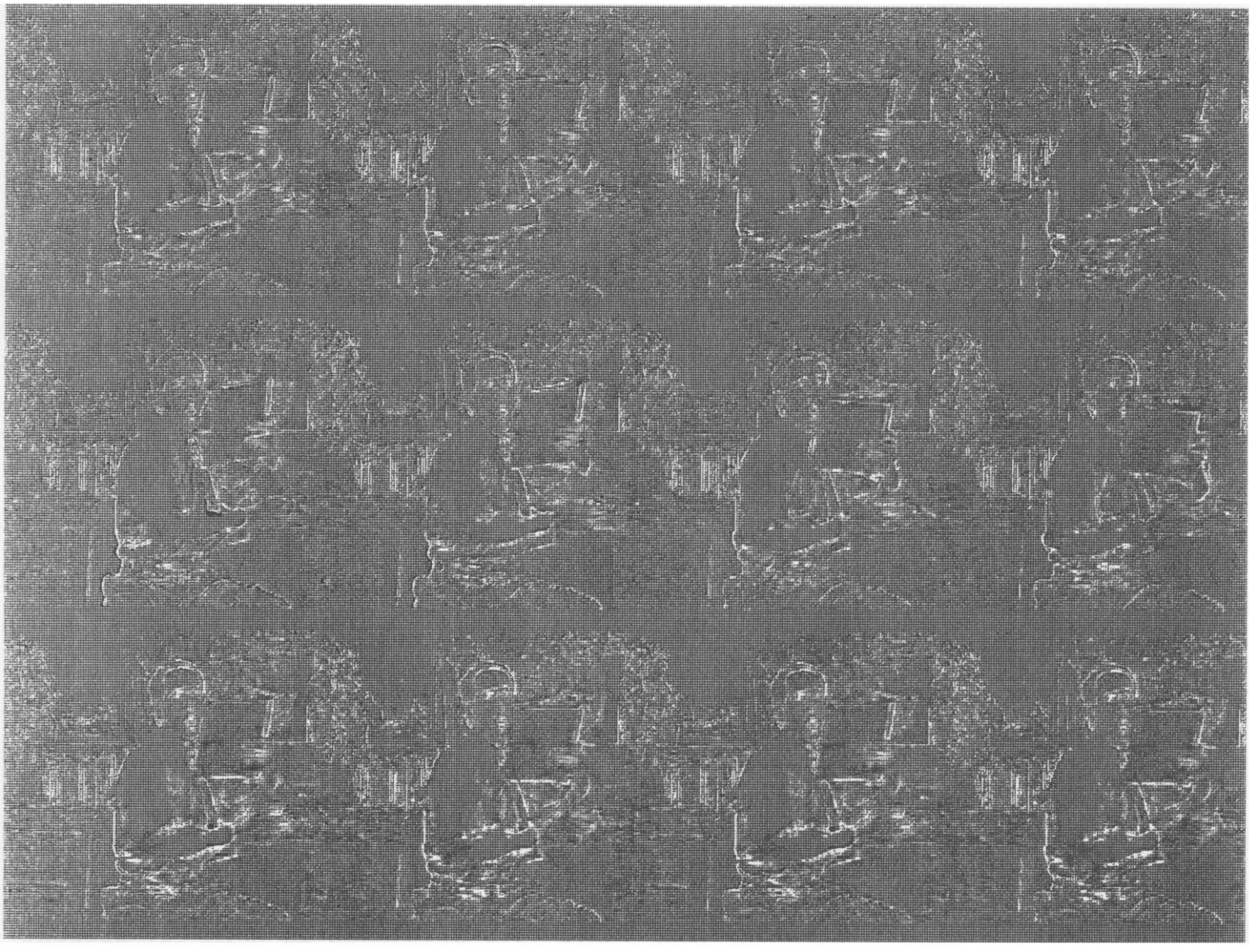

Figure 13. Sobel difference imagery of Figure 7. Top row - rate DS1. Second row - rate 1/2 DS1. Third row - rate 1/4 DS1.

Table 6. Spatial Blurring Features For Figure 13

Scene	M-PSDI	SD-PSDI	RMS-PSDI	NPGT-PSDI
Top Row (DS1)	17.5	35.2	39.4	5443
Second Row (1/2 DS1)	19.0	37.9	42.4	6562
Bottom Row (1/4 DS1)	22.5	47.4	52.5	9544

Table 7. False Edge Features For Figure 13

Scene	M-NSDI	SD-NSDI	RMS-NSDI	NPLT-NSDI
Top Row (DS1)	-9.0	20.7	22.6	1138
Second Row (1/2 DS1)	-9.6	22.5	24.4	1577
Bottom Row (1/4 DS1)	-11.5	25.0	27.5	1989

2.7 Jerkiness Feature Using Position Errors

Jerkiness is a video teleconferencing/telephony artifact in which the original smooth and continuous imagery motion is perceived as a series of distinct snapshots at the output (see Table 1). Jerkiness is normally present when a codec data compression algorithm achieves data compression by elimination of fields or frames. The number of fields and/or frames that are eliminated (not transmitted) is not necessarily guaranteed to be an accurate measure of jerkiness. Sophisticated coding algorithms can update different portions of the image at different frame rates and even interpolate missing frames to achieve smooth motion effects. Jerkiness is present when the position of a moving object within the video scene is not updated rapidly enough. Section 2.7.1 proposes a measure for jerkiness based on injecting a video scene containing a moving object, and then measuring the object's position

errors in the output video. The technique is general enough to use an arbitrary object which is undergoing translational motion. A stored image of the stationary object is required to implement the technique. Although the jerkiness feature presented in section 2.7.1 is very accurate, there is one shortcoming. The feature cannot, in general, be extracted from arbitrary video scenes. Section 2.8.1 of this report will propose another measure of jerkiness which can be extracted from any video scene.

For moving objects, the proposed measure of jerkiness complements the previously proposed measures of spatial blurring in that the jerkiness feature measures temporal positioning accuracy of the object while the spatial blurring features measure the spatial resolution of the object. Data compression of motion video often involves a tradeoff between allocating bits to the temporal or spatial attributes of moving objects. The ability to measure separately the temporal and spatial attributes raises the possibility of tailoring performance specifications to the application. For example, consider the application of VTC/VT for trouble shooting circuit diagrams. In this application, high spatial resolution of the circuit diagram (assumed to be mostly stationary) is much more important than having a moving pointer (such as a finger or pen) seen as smooth and continuous. In other applications involving head and shoulders video teleconferencing, having less jerkiness may be more important than having high spatial resolution.

2.7.1 Feature Extraction Technique

A feature for estimating the jerkiness in sampled video imagery can be obtained using very simple image processing techniques. The jerkiness feature can be extracted by injecting a video scene that contains a moving object. The horizontal and vertical motion of the object is then tracked for the output imagery. Comparing the vertical and horizontal motion trajectories of the output to the input, a useful measure of jerkiness is obtained. The input motion trajectory can be obtained from processing the input video scene, or from a priori knowledge (since the test signal is known). In this manner, the amount of jerkiness in the output imagery relative to the input imagery can be determined. Several

steps are required to apply the technique. Time alignment of the input and output video scenes before processing is not required. The alignment will be performed on the input and output motion paths (see step 5), rather than on the input and output video scenes. The following steps are applied to extract the feature.

1. Stationary reference object

   A stationary reference image of an object against a uniform background is stored. This reference image of the object will be used to track motion jerkiness. The technique is general so that any non-rotational, non-growing or shrinking object may be used. For simplicity, a black ball on a white background was used for the experiments presented later in this report.

2. Moving reference video scene

   Successive frames of the object in step 1 above are generated with the object moving (translating in vertical and horizontal positions). The object may be moved horizontally, vertically, or diagonally depending upon whether one desires to test the jerkiness in the horizontal, vertical, or diagonal directions. The object may also be moved at different velocities to test the jerkiness over a wide range of motion in the video scene. In this manner, a video scene is generated that contains an object moving according to some known motion path (the vertical and horizontal positions of the object are known for each video frame).

3. Output video scene

   The generated video scene from step 2 above is injected as the test signal. The output video scene is recorded or frame grabbed into the video quality assessment system. For greater accuracy, each field (1/60th of a second) was recorded for the experiments presented later in this report.

4. Output motion path

The vertical and horizontal positions of the moving object are obtained by correlating (see Oppenheim and Schafer, 1975) each video frame of the output video scene (from step 3 above) with the reference object (from step 1 above). In this manner, the vertical and horizontal motion paths of the moving object are found for the entire output video scene. Correlation yields a very robust and accurate estimate of the moving object's position. However, correlation is also computationally expensive. A computationally more efficient, but less accurate, method of tracking the moving object's position is available if the object is against a black background. Then one could obtain the object's motion path by computing the centroid of the object for each video frame (Tzafestas, 1986). Nevertheless, the correlation method was used for the experiments presented later in this report.

5. Aligned output motion path

The output motion path of the object (from step 4) is aligned with the true motion path (from step 2). Alignment of the input and output motion paths is required to compensate for absolute video delay of the device under test. The alignment procedure used here corresponds to what a viewer would observe if that viewer were insensitive to the absolute video delay. The best alignment of the output motion path to the input motion path is simply that which produces the smallest average sum of the squared vertical and horizontal position errors (the sum of the squared position errors is first performed over all frames of the video scene, then this sum is divided by the number of frames in the video scene). The jerkiness feature is then calculated as the square root of this average sum of the squared position errors. A mathematical definition for this jerkiness feature, henceforth called temporal root mean square position error (TRMS-PE), is given in equation 15 of Appendix A.

2.7.2 Sample VTC/VT Results

The notion of testing the jerkiness of motion video first occurred when the output of a VTC/VT codec was monitored at bit rates on the order of DS1. An object that moved across the field of view of the camera did not seem to move as smoothly after the scene had passed through the codec. A quantification of how jerky the distortion mechanism was and how it varied with code rate and speed of the object was sought.

An ideal test signal for jerkiness would be a computer generated scene of an object moving at a constant speed across the screen at a specified angle (horizontally, vertically, diagonally). Due to equipment limitations, test scenes were generated using a black ball suspended by a long pendulum (about 15 feet) against a backlit (white) background. Since only a small portion of the center part of the swing was used, the ball's speed and angle were approximately constant.

A black ring was placed on the backlit background so that background movement due to imperfections in the test setup or recorders could be detected and taken into account. For very stable recorders or computer generated scenes, the black ring would not be necessary.

To generate test scenes of different speeds, the ball was dropped from different heights. To generate test scenes at different angles, the camera was tilted to the appropriate angle. In this manner, test scenes were generated for horizontal and 45 degree angles at several different velocities (ball heights). Three consecutive swings from each ball height were captured into the computer. For each scene, every set of two fields that could be displayed on the video cassette recorder in still frame mode was captured and stored in a file for later processing. Images were grabbed for every NTSC field increment of the recorder (1/60 second). Although the speed of the consecutive swings for each ball height was slightly decreasing, the motive was to establish the accuracy and repeatability of the jerkiness measurement by examining three independent trials at each ball speed. The following scenes with horizontal motion were captured into the computer and analyzed:

1. Nine original reference scenes (three consecutive swings of the ball for each of three different ball heights or speeds).

2. Nine degraded codec output scenes at the ball's fastest speed (DS1, 1/2 DS1, and 1/4 DS1 code rates for the three consecutive swings at the fastest speed).

3. Six degraded codec output scenes at the ball's medium speed (DS1, and 1/4 DS1 code rates for the three consecutive swings at the medium speed).

4. Six degraded codec output scenes at the ball's slowest speed (DS1, and 1/4 DS1 code rates for the three consecutive swings at the slowest speed).

The following scenes with 45 degree diagonal motion were captured and analyzed:

1. Three original reference scenes (three consecutive swings of the ball at the fastest speed).

2. Nine degraded codec output scenes at the ball's fastest speed (DS1, 1/2 DS1, and 1/4 DS1 code rates for the three consecutive swings of the ball at the fastest speed).

The horizontal and vertical motion paths of the ball for each scene listed above were obtained by correlating a stored reference ball with each image of the video scene. Possible movement of the background (which contained a black ring) due to imperfection in the test setup was detected by correlating a stored reference ring with each image of the video scene. The motion of the background in the test setup was found to be on the order of one or two pixels and hence was neglected.

Figure 14 shows four sequential images grabbed (every 1/60th of a second from left to right) at the various bit rates for a horizontally moving ball. The top row in Figure 14 shows four consecutive field increments of the original NTSC signal, the next three rows show the corresponding codec outputs at bit rates of DS1, 1/2 DS1, and 1/4 DS1, respectively. For viewing convenience, single-frame temporal alignment has been applied to the video in Figure 14. Note that the second and third images in each row of the codec output are correctly aligned with

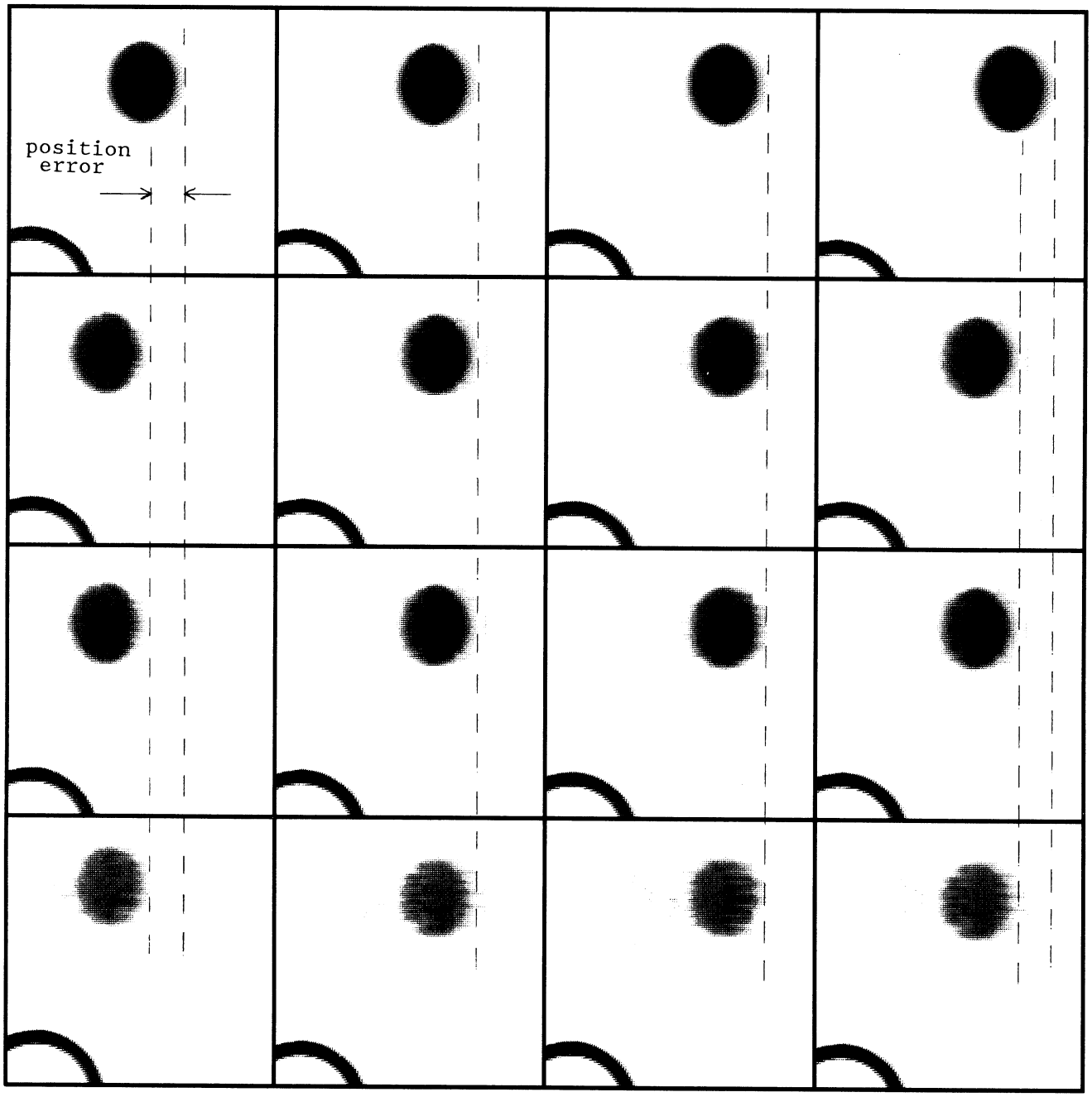

Figure 14. Four sequential VTC/VT images for a horizontally moving ball. Images were grabbed every 1/60 second from left to right. Original (first row), DS1 (second row), 1/2 DS1 (third row), 1/4 DS1 (last row).

the NTSC input. For each bit rate, the ball is identically positioned, but this positioning is not the same as the input in the first and fourth codec output images. In addition, for each bit rate, the ball in the fourth codec output image appears to have backed up while the original continues to advance from left to right. The reason for the strange positioning of the moving ball in the codec output video will be explained below. Figure 15 shows a portion of the diagonal test data with the ball moving from the upper left to the lower right. The format of Figure 15 is the same as that of Figure 14.

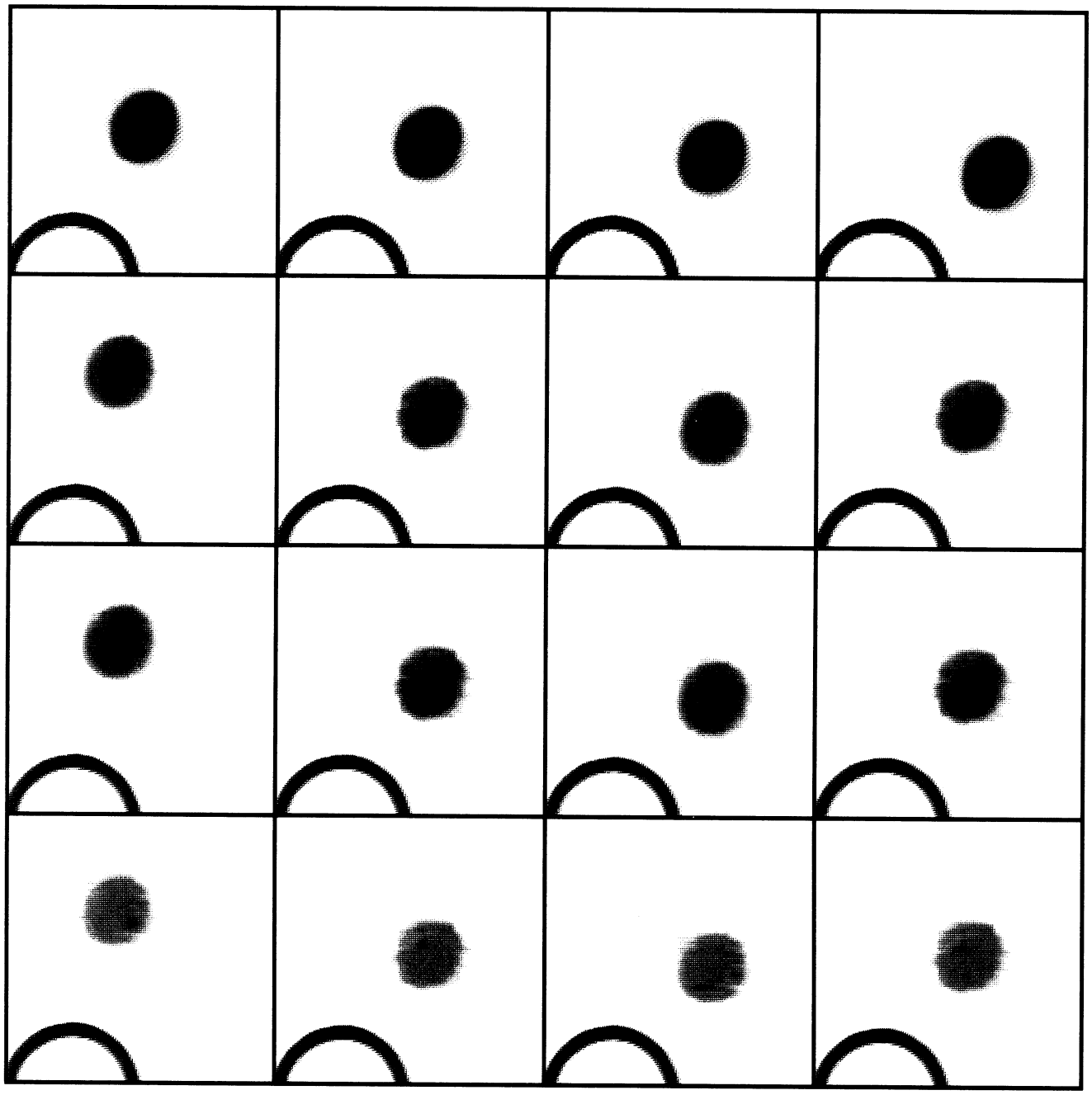

Figure 15. Four sequential VTC/VT images for a diagonally moving ball. Images were grabbed every 1/60 second from left to right. Original (first row), DS1 (second row), 1/2 DS1 (third row), and 1/4 DS1 (last row).

Figure 16 shows the horizontal and vertical positions of the ball as a function of field number for fast motion at the horizontal angle. The ball positions are plotted for the original NTSC scene and for a code rate of DS1. Figure 17 shows the ball positions for fast motion at the diagonal angle of approximately 45 degrees. In Figures 16 and 17, the codec very accurately positioned the ball for two consecutive fields, but then, to save on transmission, simply repeated these two fields before accurately placing the ball again. This omission and repetition of every other frame caused the backup mentioned earlier in Figures 14 and 15. Examining Figure 14, the ball position in the first DS1 output image corresponds to field number 3 in Figure 16. In the second DS1 output image, the ball jumps a large distance to field position 4 in Figure 16. The ball in the second and third DS1 output images was accurately placed (corresponding to field numbers 4 and 5 in Figure 16). Then, the ball in the fourth DS1 output image (field number 6 in Figure 16) backed up because the codec output the same field that occurred earlier in time (field number 4 in Figure 16). Thus, the fourth DS1 output image in Figure 14 was identical to the second DS1 output image (since field number 6 is identical to field number 4 in Figure 16).

In order to measure the TRMS-PE feature, the input and output motion paths had to be aligned according to processing step 5 of section 2.7.1. Figure 18 shows the aligned motion paths for the diagonal case in Figure 17 that minimizes the root mean square position error. The TRMS-PE feature can be calculated from equation 15 of Appendix A.

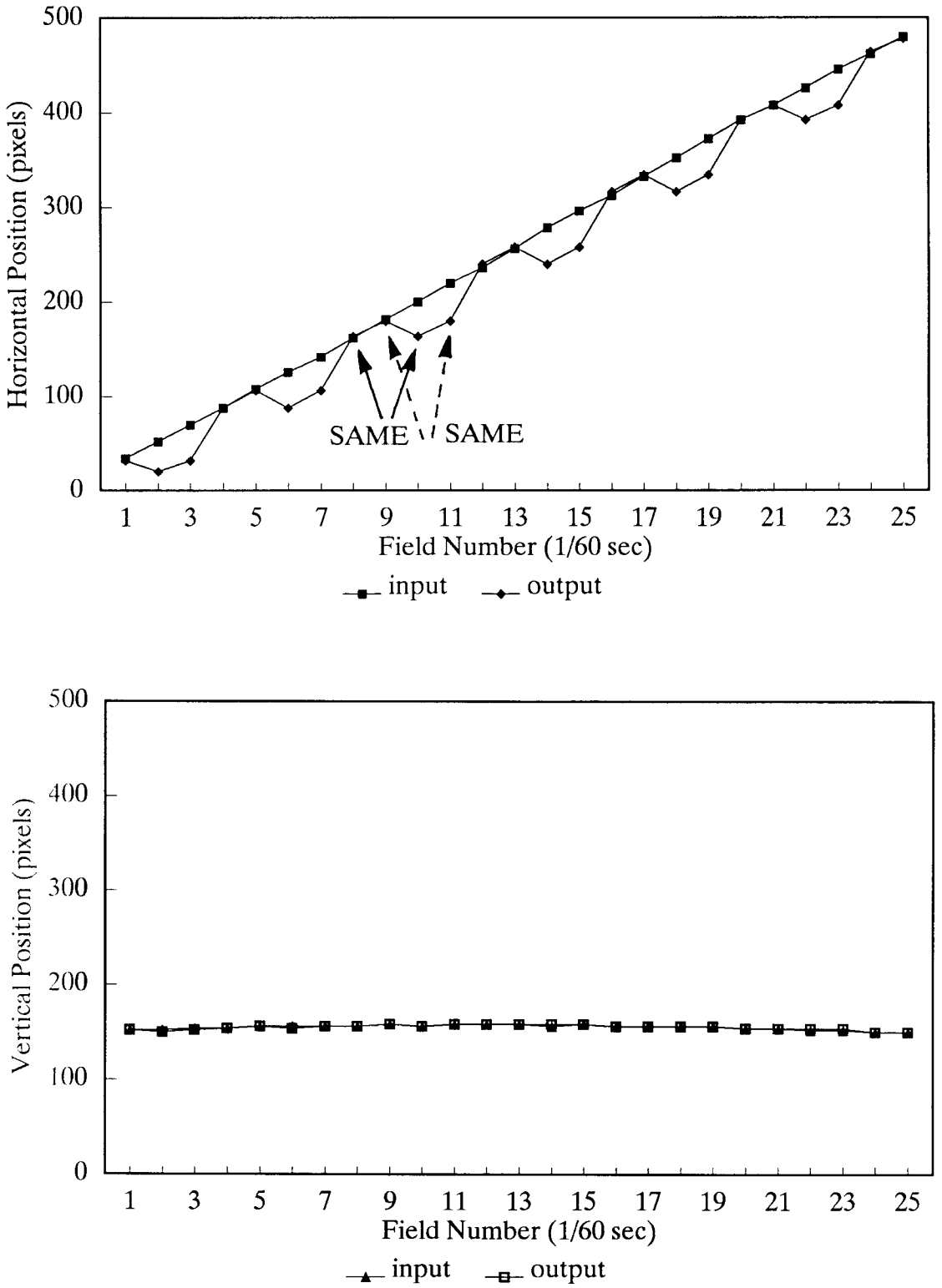

Figure 16. Positions of moving ball as a function of field number for fast motion at the horizontal angle. Ball positions plotted for original NTSC scene and for a code rate of DS1.

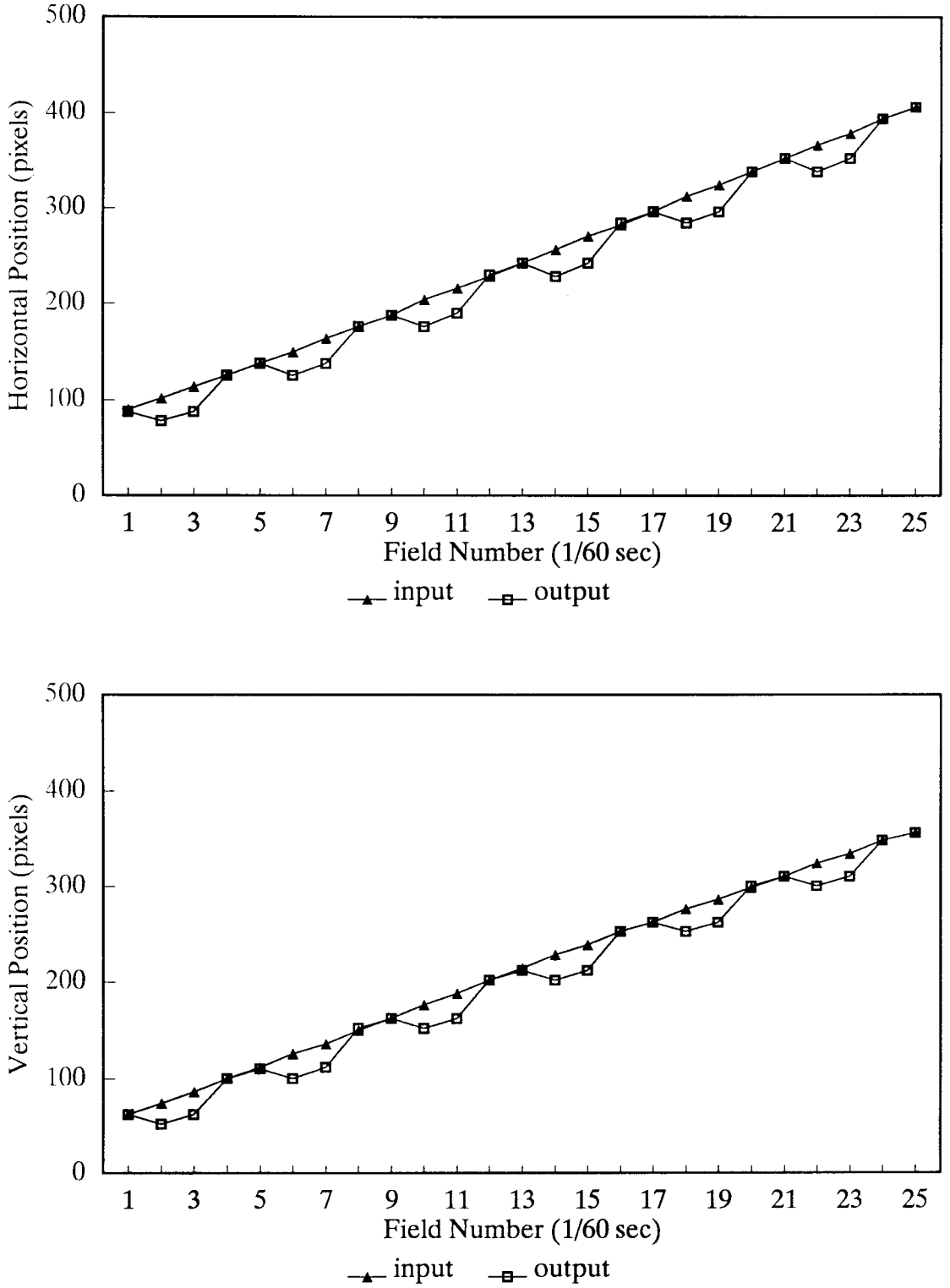

Figure 17. Positions of moving ball as a function of field number for fast motion at the diagonal angle. Ball positions plotted for original NTSC scene and for a code rate of DS1.

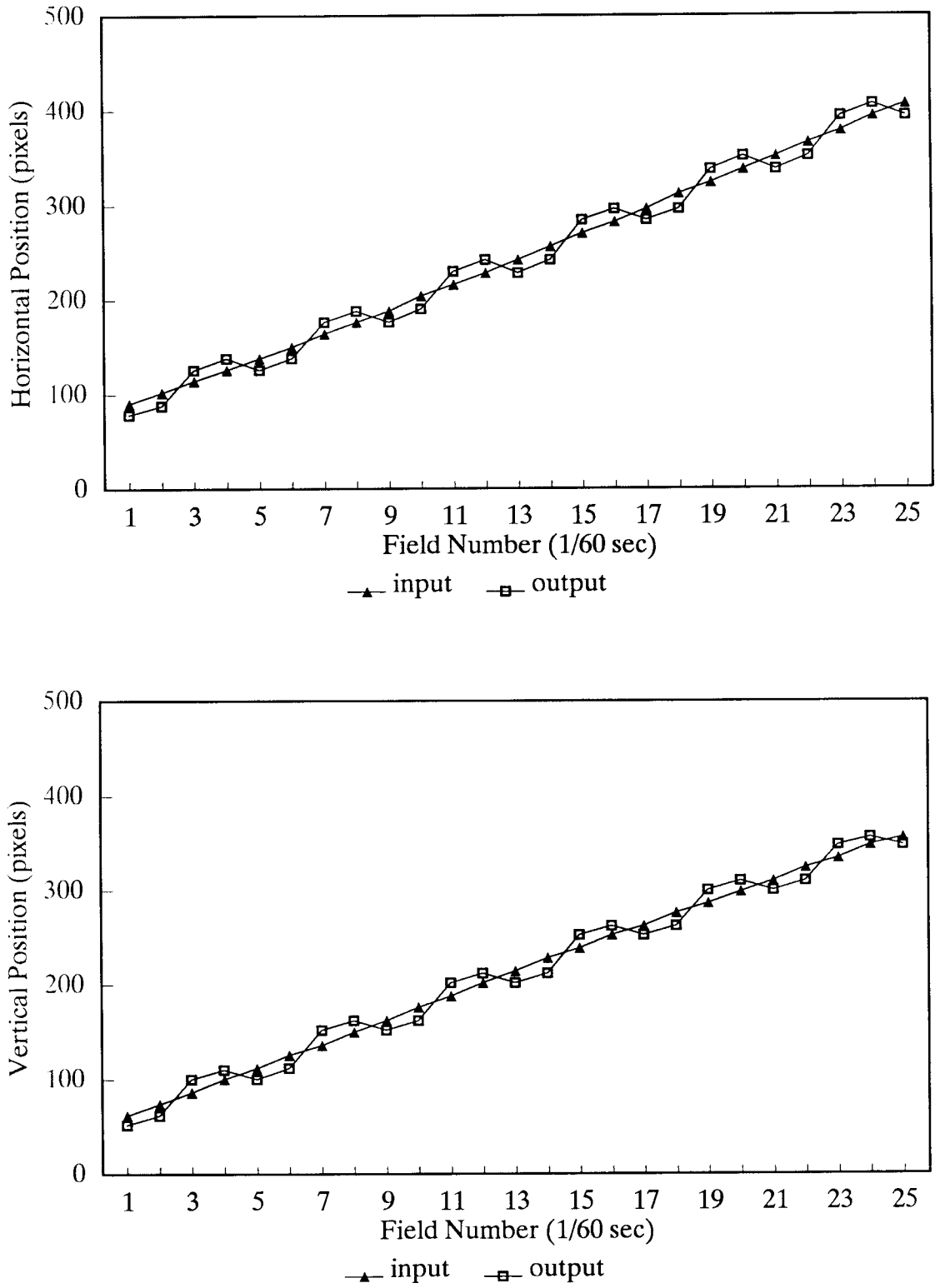

Figure 18. The aligned motion paths for the diagonal case in Figure 17.

The same number of fields in the middle portion of the motion paths was used to calculate the TRMS-PE feature for each trial within each test case. Each input path speed was found by taking the difference in pixels between the endpoints of the middle portion of the motion path divided by the total number of fields (thus, speed is measured in pixels per field). Table 8 summarizes the TRMS-PE results for all the test cases mentioned previously. In Figure 19, the TRMS-PE is plotted verses the speed of the ball for two particular code rates (1/4 DS1, and DS1) and a particular type of motion (horizontal motion). The TRMS-PE feature reflects how far off the output object positions are from the input object positions, on the average. For faster speeds, the output positions are proportionally farther off from the input positions. Thus, the TRMS-PE feature is also proportionally higher. The plot in Figure 19 shows the linear variation of TRMS-PE with speed as described above. In Figure 20, the TRMS-PE is plotted verses the code rate for a particular speed group (fast) and a particular type of motion (diagonal). Here, three trials are shown for each code rate. Since each trial is slightly slower than the previous (three consecutive swings of the ball), there is a slight variation in TRMS-PE between the trials. There is no variation in TRMS-PE with code rate, so the codec is not changing the location to which the output object is placed. The codec is only changing the amount of spatial resolution it allocates to the object (see Figures 14 and 15).

Table 8. Summary Of TRMS-PE Results

Orientation	Code Rate	Speed (Pixels/Field)	TRMS-PE
horizontal	1/4 DS1	17.79	19.13
horizontal	1/4 DS1	17.26	19.01
horizontal	1/4 DS1	16.95	18.13
horizontal	1/4 DS1	9.89	10.84
horizontal	1/4 DS1	9.58	10.86
horizontal	1/4 DS1	9.47	10.51
horizontal	1/4 DS1	6.53	7.91
horizontal	1/4 DS1	6.42	7.51
horizontal	1/4 DS1	6.21	7.08
horizontal	1/2 DS1	17.79	19.09
horizontal	1/2 DS1	17.26	18.88
horizontal	1/2 DS1	16.95	18.13
horizontal	DS1	17.79	19.10
horizontal	DS1	17.26	18.76
horizontal	DS1	16.95	18.01
horizontal	DS1	9.89	10.60
horizontal	DS1	9.58	10.28
horizontal	DS1	9.47	10.15
horizontal	DS1	6.53	7.40
horizontal	DS1	6.42	7.20
horizontal	DS1	6.21	6.70
diagonal	1/4 DS1	17.28	18.49
diagonal	1/4 DS1	16.99	17.93
diagonal	1/4 DS1	16.39	17.71
diagonal	1/2 DS1	17.28	18.54
diagonal	1/2 DS1	16.99	18.19
diagonal	1/2 DS1	16.39	17.52
diagonal	DS1	17.28	18.47
diagonal	DS1	16.99	18.31
diagonal	DS1	16.39	17.88

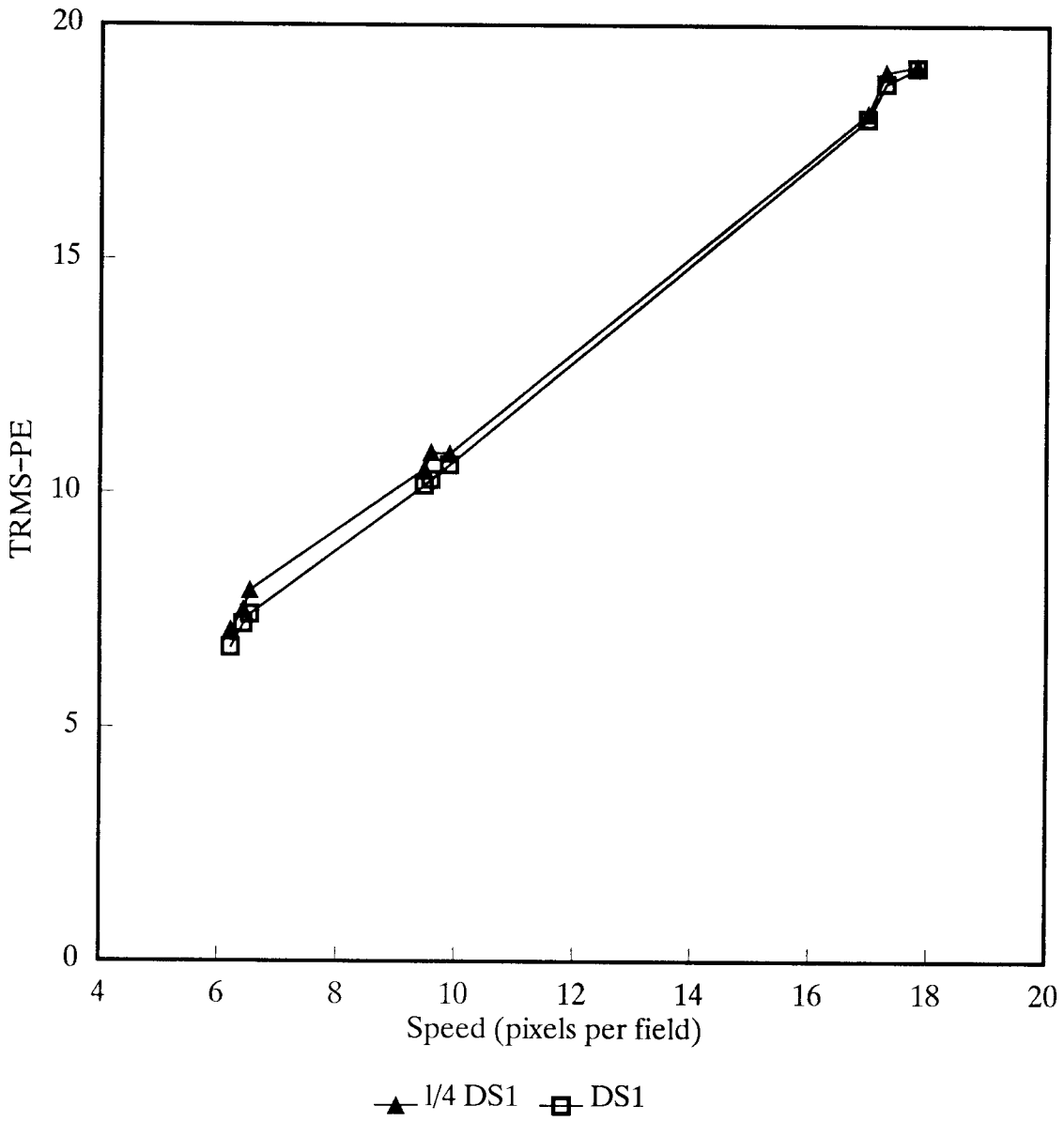

Figure 19. TRMS-PE plotted as a function of horizontal ball speed for code rates of 1/4 DS1 and DS1.

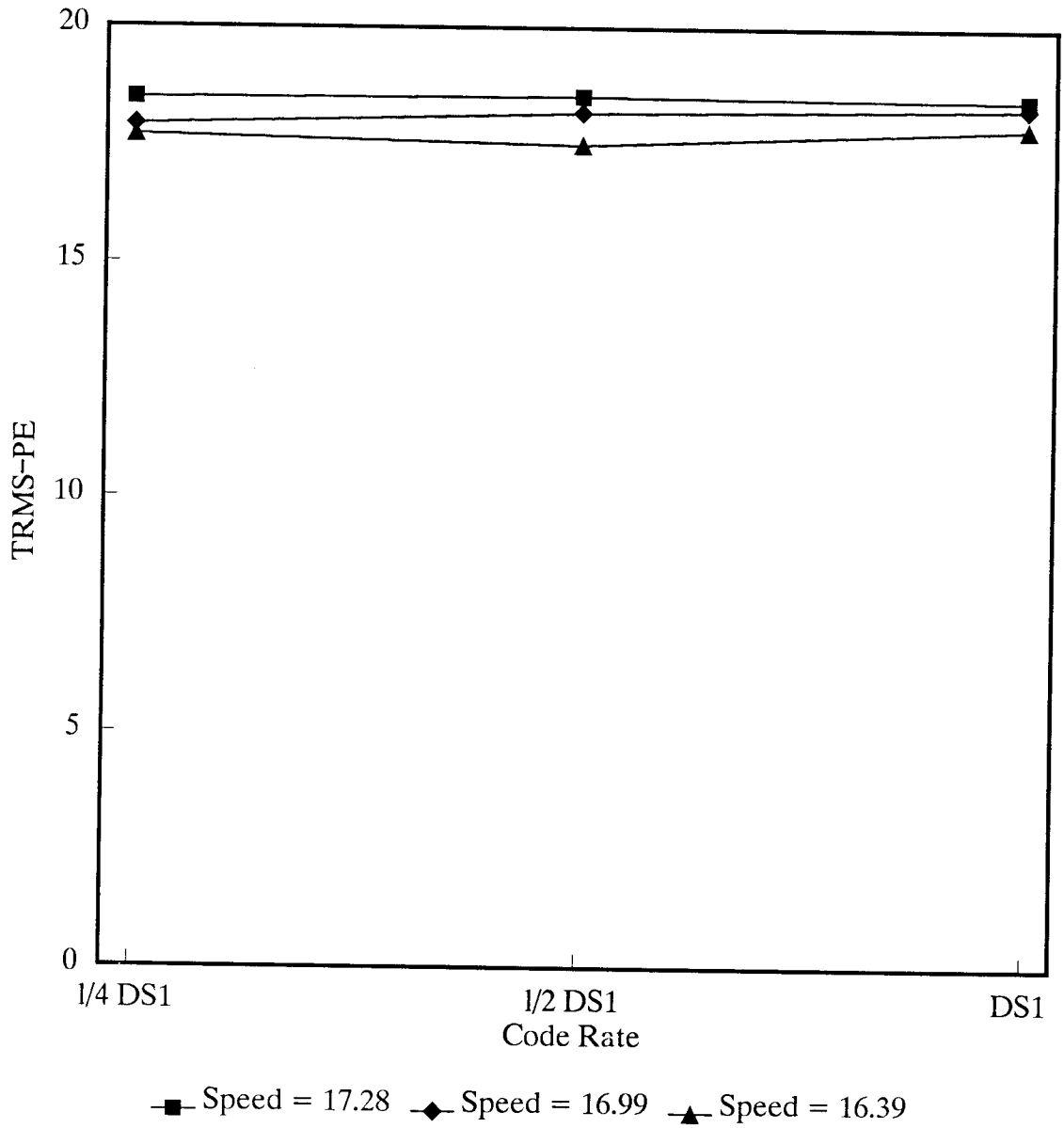

Figure 20. TRMS-PE plotted as a function of code rate for the fast speed group and diagonal motion. Three trials with slightly different speeds are shown.

Applying the TRMS-PE measure of jerkiness to one particular codec has illustrated several important insights into the operation of VTC/VT. First, the TRMS-PE jerkiness measure is very stable; in fact, the small variation in speed between consecutive swings of the ball shows up as a small variation in TRMS-PE, as expected. Second, the jerkiness, or temporal update of the codec, may not vary with code rate (as in Figure 20). The particular codec tested here achieved bit reduction by degrading the spatial resolution of scenes and not the frequency of update. For the particular codec tested, the jerkiness was due to omission of every other frame, regardless of operating bit rate. This simple result would not necessarily have been obtained for codecs that use more sophisticated coding/decoding methods. Other codec algorithms for attaining less jerkiness might trade spatial resolution for more or less temporal positioning accuracy.

2.8 Jerkiness Feature Using Difference Image

The TRMS-PE measure of jerkiness cannot easily be applied to arbitrary video scenes. Section 2.8.1 proposes a measure of jerkiness that can be applied to any video scene. The genesis of this new measure of jerkiness occurred when observing codec input and output video that had been aligned using the single-frame temporal alignment method discussed earlier (as in Figure 14). If one were to compute the difference images of the input and the output video, image pairs that contained no positioning errors (second and third images of each row in Figure 14) would yield smaller difference errors than image pairs that contained positioning errors (first and fourth images of each row in Figure 14). As a function of time, the total composite difference error of a moving object would be composed of two components. One component represents errors due to blurring or distortion of the object. The other component represents errors due to incorrect positioning of the object.

Section 2.8.1 presents a method for extraction of three new features. One of the features will be shown to be intimately related to jerkiness. The other two features represent the average distortion of the output video due to jerkiness and spatial blurring. The exact feature extraction technique and sample VTC/VT results are discussed in detail next.

2.8.1 Feature Extraction Technique

The features are extracted from the undistorted input and distorted output sampled video. The feature extraction technique is computationally efficient and possesses many of the other desirable properties of features that were previously mentioned. The features are extracted from the standard deviation of the difference images (input image minus output image), where the input and output video has been time aligned using the single-frame temporal alignment method. The standard deviation of the difference image is used, instead of the mean or root mean square, because the standard deviation is insensitive to gray level shifts in the sampled video. The exact feature extraction method follows:

1. Video alignment

   Single-frame temporal alignment of the input and output video is performed.

2. Difference image

   For each aligned video image pair, a difference image is formed by subtracting the output image from the input image.

3. Standard deviation of the difference image (SD-DI)

   The standard deviation of each difference image (SD-DI), from step 2 above, is computed as the square root of (the summation of the squares of the image pixel values divided by the total number of pixels, minus the square of the mean of the difference image). Here, the mean of the difference image is computed as the summation of the image pixel values divided by the total number of pixels. See equation 16 in Appendix A for a mathematical definition of SD-DI.

4. Feature computation

   From the time history of SD-DI, from step 3 above, the following three features are computed:

   a. The temporal mean of SD-DI (TM-SD-DI)

      TM-SD-DI is computed as the summation of the SD-DI values divided by the total number of SD-DI values. TM-

SD-DI is primarily related to the average distortion caused by spatial blurring and jerkiness. See equation 17 in Appendix A for a mathematical definition of TM-SD-DI.

b. The temporal standard deviation of SD-DI (TSD-SD-DI)
TSD-SD-DI is computed as the square root of (the summation of the squares of the SD-DI values divided by the total number of SD-DI values, minus the square of TM-SD-DI). This estimate of the standard deviation of the population of SD-DI time samples is asymptotically unbiased for a large number of SD-DI values. An alternate method of computing TSD-SD-DI that is unbiased for a small number of SD-DI values may be used instead (see, for example, Crow et al., 1960). TSD-SD-DI is primarily related to jerkiness. See equation 18 in Appendix A for a mathematical definition of TSD-SD-DI.

c. The temporal root mean square of SD-DI (TRMS-SD-DI)
TRMS-SD-DI is computed as the square root of (the summation of the squares of the SD-DI values divided by the total number of SD-DI values). TRMS-SD-DI is related to the total distortion caused by spatial blurring and jerkiness. See equation 19 in Appendix A for a mathematical definition of TRMS-SD-DI.

To compute the amount of distortion in the difference images with respect to the input images, the SD-DI values could be normalized by the standard deviation of the undistorted input video. Alternatively, normalized features could be obtained by dividing TM-SD-SD, TSD-SD-DI, and TRMS-SD-DI by the temporal mean of the standard deviation of the undistorted input video. Thus, normalized features closer to zero will represent smaller distortions while normalized features closer to one will represent larger distortions.

2.8.2 Sample VTC/VT Results

The VTC/VT imagery of Figure 7 was processed to extract the TM-SD-DI, TSD-SD-DI, and TRMS-SD-DI features as described above. The difference images were obtained by subtracting the output images (rows two, three, and four of Figure 7) from the single-frame temporally aligned input images (row one of Figure 7). Figure 21 shows the resulting difference images, where rows one, two, and three of Figure 7 correspond to codec bit rates of DS1, 1/2 DS1, and 1/4 DS1, respectively. For display purposes only, the difference images of Figure 21 have been scaled such that gray (intensity of 128) represents no error, black (intensities from 0 to 127) represents negative error, and white (intensities from 129 to 255) represents positive error. Note that images 1 and 2 (from left to right) for codec bit rates DS1 and 1/4 DS1 contain small errors, while images 3 and 4 for a codec bit rate of 1/2 DS1 contain small errors. The particular codec under test achieved some of its data compression by discarding every other input frame and repeating every other decoded output frame (one frame represents two images in Figure 7 since the images were grabbed for each field increment of the video recorder, and there are two fields for each NTSC frame). Since the input video was injected asynchronously for each of the three codec bit rates, there was no guarantee that the same frames would be discarded for all bit rates. In Figure 14, the same video frames were discarded for all bit rates (by chance) while in Figure 7 the same frames were discarded for the DS1 and 1/4 DS1 bit rates but different frames were discarded for the 1/2 DS1 bit rate. Thus, care should be taken to process a sufficiently long time sequence of images when extracting the TM-SD-DI, TSD-SD-DI, and TRMS-SD-DI features. Otherwise, inaccurate results may be obtained, particularly for codecs that discard a large number of video frames.

The first two difference images for rate DS1 in Figure 21 contain errors due to blurring. The third and fourth difference images contain errors due to blurring and jerkiness. Examining Figure 7 closely, each image in the NTSC video scene (top row) is unique and shows steady motion of the report crossing the man's face. Meanwhile, the third and fourth codec output images for rate DS1 (second row) are the same as the first and second codec output images, respectively. One can see from Figure 7 that the codec is performing frame repetition. Thus, on the repeated

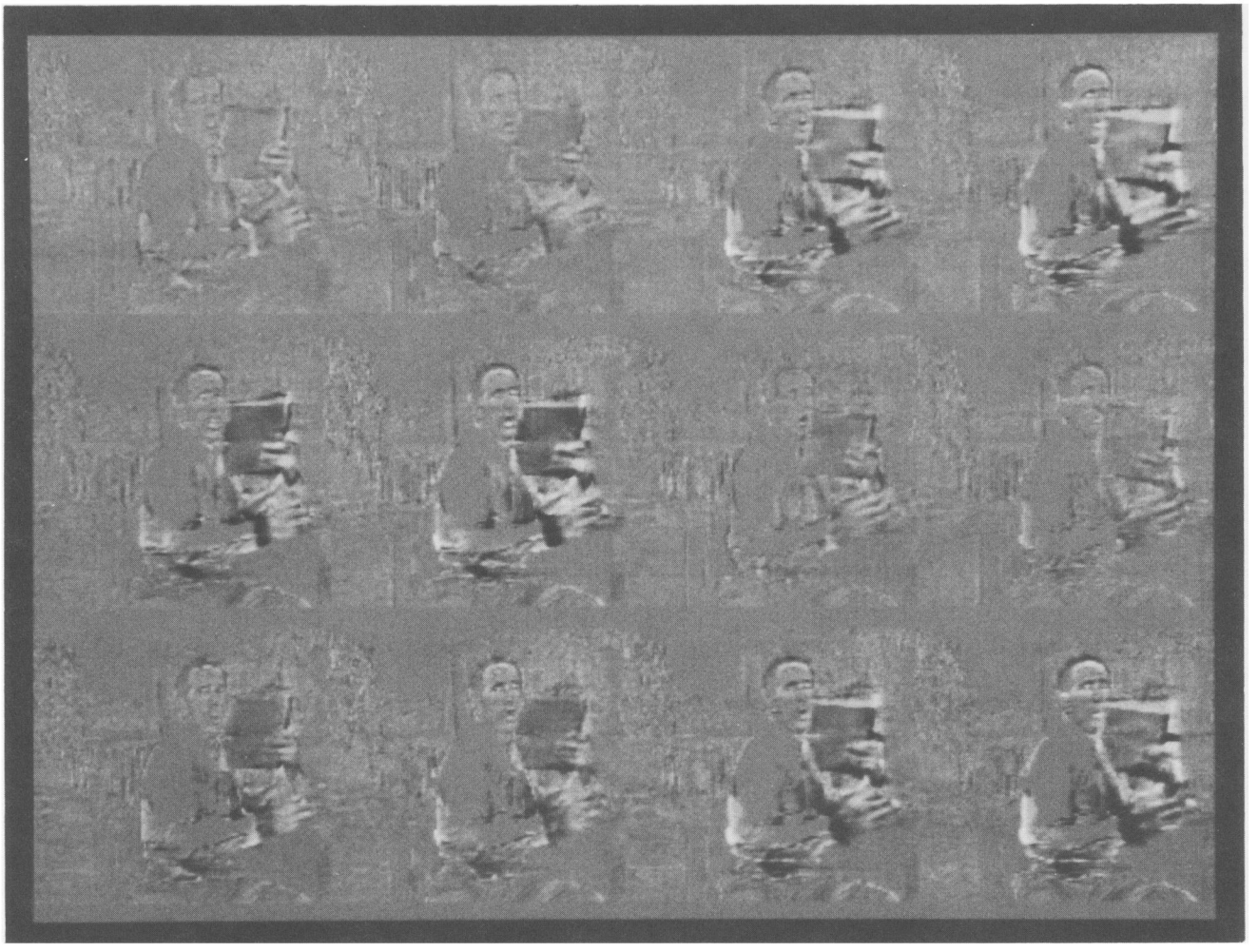

Figure 21. Difference images for VTC/VT imagery of Figure 7. Codec bit rates of DS1 (top row), 1/2 DS1 (second row), and 1/4 DS1 (bottom row).

frame (consisting of images 3 and 4 in Figure 7), large difference errors are obtained and these errors are due to jerkiness in the codec output.

Figure 22 is a graph of the time history of the SD-DI values for the images in Figure 21. The SD-DI values for the first four fields in Figure 22 were calculated from the first four difference images in Figure 21. The ball test scenes (Figures 14 and 15) and the man test scene (Figure 7) were obtained from the same codec. The time history of SD-DI very much resembles the codec output ball position errors (compare with the output ball position error with respect to the true input ball position in Figures 18).

Table 9 presents the computation of the unnormalized TM-SD-DI, TSD-SD-DI, and TRMS-SD-DI features for the eight fields of Figure 22 (for reference, the temporal mean of the standard deviation of the undistorted input video was 77.6). The temporal mean of SD-DI (TM-SD-DI) and the temporal root mean square of SD-DI (TRMS-SD-DI) represents the average and total distortion due to blurring and jerkiness. The temporal standard deviation of SD-DI (TSD-SD-DI) represents the extent of the variation of SD-DI about its mean. More jerky motion will result in larger values of TSD-SD-DI. Curiously, from Table 9, TSD-SD-DI increases slightly with increasing bit rate. This contradicts the earlier TRMS-PE measure of jerkiness which showed that jerkiness was the same for all bit rates (see Figure 20). An explanation for the phenomenon is as follows. The added spatial blurring for low bit rates versus higher bit rates tends to raise the SD-DI curve (increasing TM-SD-DI in Table 9). In the raised SD-DI curve, smaller increases in difference errors due to positioning are obtained, and this results in a flattening of the SD-DI curve (decreasing TSD-SD-DI in Table 9). Subjectively, the TSD-SD-DI measure of jerkiness may be more accurate than the TRMS-PE measure of jerkiness because added spatial blurring tends to reduce the effect of jerkiness. If the object is badly blurred, one cannot tell if the motion is jerky. If the object is focused, one readily notices jerky motion.

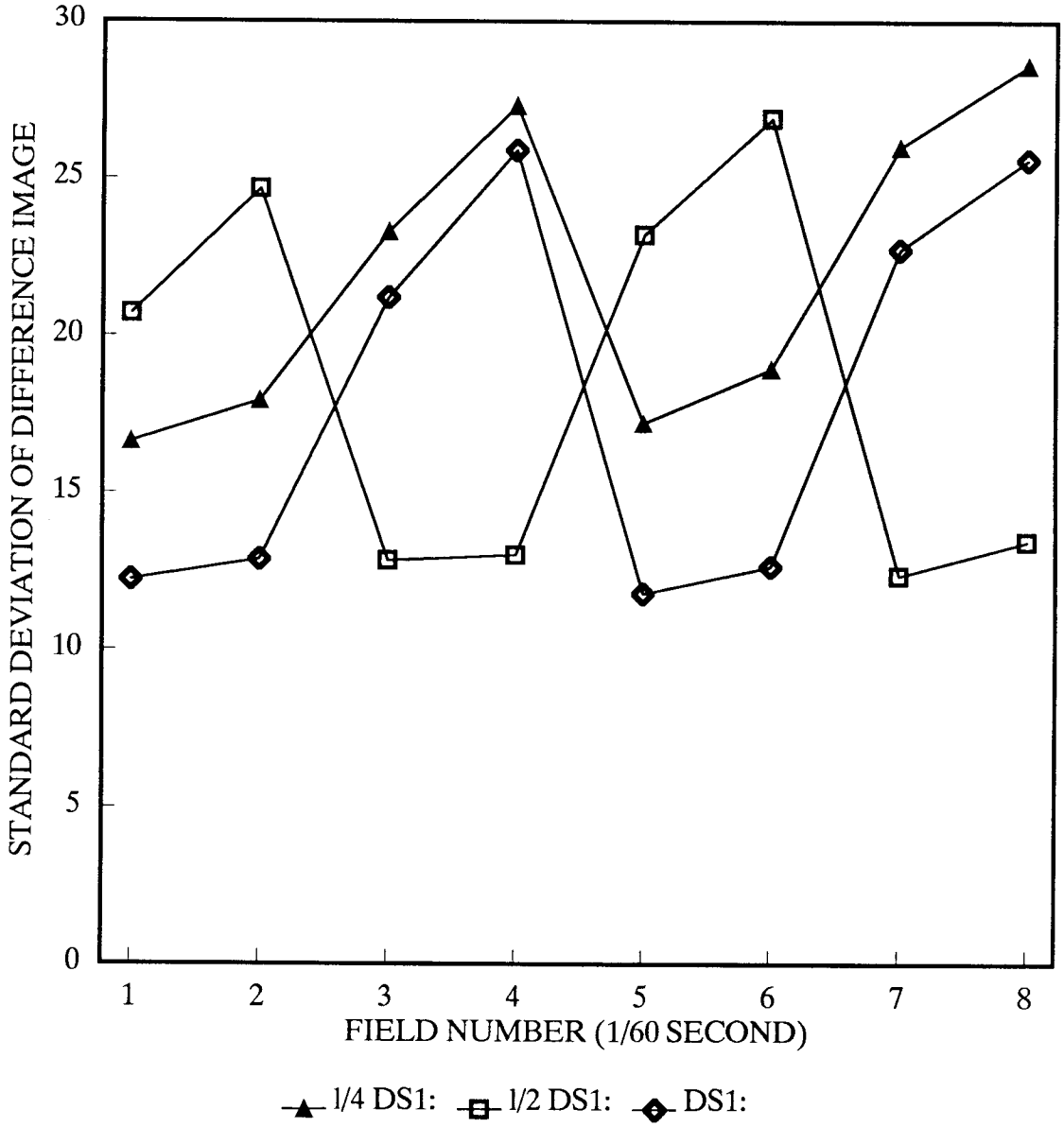

Figure 22. Time history of SD-DI for the difference images of Figure 21. The first four field numbers correspond to the four images in Figure 21.

Table 9. Summary Of SD-DI Features For Figure 22

Scene	TM-SD-DI	TSD-SD-DI	TRMS-SD-DI
DS1	18.1	5.9	19.1
1/2 DS1	18.4	5.7	19.3
1/4 DS1	22.0	4.6	22.5

3. CONCLUSIONS AND RECOMMENDATIONS

Objective feature extraction techniques have been presented that measure the predominant artifacts present in digitally transmitted video systems. Among these artifacts are blurring/smearing, blocking, edge busyness, image persistence, and jerkiness. Features are extracted from the digitized video imagery that reflect degradations perceived by the viewer. The features are sensitive to the type of video being transmitted which is important since the performance of digital codecs depend strongly on the type of video being transmitted. In addition, the features possess many of the desirable properties that humans also possess, including the potential adaptability to focus attention on local disturbances in the video. Thus, the features are expected to correlate strongly with subjective quality ratings.

Depending upon the feature one wishes to extract, the method for temporally aligning the input and distorted output video frames varies. Spatial blurring and jerkiness measures have been presented that do not require the input and output video scenes to be aligned. Other measures, such as edge busyness, blocking, image persistence, and jerkiness for natural motion scenes, require some form of temporal alignment. Two possible methods of temporally aligning the input and output video were presented. The computational requirements of the proposed features varied. However, these computational requirements appear reasonable for modern digital signal processing systems.

Spatial blurring features were presented that relate to the sharpness of the edges in the video imagery. These spatial blurring features appear to be applicable to many types of video imagery, including natural scenes. Blocking, edge busyness, and image persistence were shown to be forms of false edge energy appearing in the output

video. Since the importance of edges are well recognized in the areas of human vision and object recognition, it is expected that these spatial blurring, blocking, edge busyness, and image persistence features will correlate especially well with subjective quality ratings.

Two measurement techniques for the jerkiness artifact of digitally transmitted video systems have been proposed. The temporal root mean square position error (TRMS-PE) feature compares the horizontal and vertical positions of a moving object in the output video scene to those in the input video scene. Although a ball was used for the moving object in the presented example, the technique is general enough to substitute any object. The TRMS-PE feature has been shown to be an accurate and repeatable measurement that determines the temporal positioning accuracy of a codec. Additional features were presented that measured the jerkiness of arbitrary video scenes. These included the missing frame ratio (MFR) feature and the TSD-SD-DI feature computed from the standard deviation of the error difference images. Video data compression is often a tradeoff between allocating bits between temporal positioning accuracy and spatial resolution. The ability to measure separately these two attributes raises the possibility of tailoring performance specifications to the application.

Further work needs to be done to determine the optimal method for combining all of the extracted feature values to produce an overall quality rating (the quality classification subsystem shown in Figure 1). Properly combining the many feature measurements into an overall quality assessment rating may require an understanding of the temporal and spatial properties of the eye and brain. To be universally useful, the quality classification subsystem must perform well over a wide range of applications. To obtain this goal, the quality classification system may require user specific application information. Subjective test results on imagery that spans the full range of digitally transmitted video systems should be used to select an optimal set of features, to train the quality classification subsystem, and to evaluate the performance of the completed system.

4. ACKNOWLEDGEMENTS

The author would like to thank Tim W. Butler, Keith E. Junker, Margaret H. Morris, and Dara Parsavand at the Institute For Telecommunication Sciences for writing image processing software that was used to produce images in this report. Further credit and appreciation is extended to Dara Parsavand for his innovative work in developing and testing the temporal root mean square position error (TRMS-PE) feature. Thanks are also extended to Edmund A. Quincy at the Institute For Telecommunication Sciences for the use of video equipment that was required to produce the test data presented in this report.

5. REFERENCES

Barten, P. G. L. (1988), Evaluation of CRT displays with the SQRI method, Society for Information Display International Symposium Digest of Technical Papers, v. XIX, May 24-26, pp. 445-448.

Barten, P. G. L. (1987), The SQRI method: A new method for the evaluation of visible resolution on a display, Proceedings of the Society for Information Display, v. 28/3, pp. 253-262.

Biberman, L. M. (1973), Perception of Displayed Information, Plenum Press, New York-London, pp. 87-119.

Biederman, I. (1985), Human image understanding: recent research and a theory, Computer Vision, Graphics, and Image Processing, v. 32, pp. 29-73.

Carlson, C. R., and R. W. Cohen (1980), A simple psychophysical model for predicting the visibility of displayed information, Proceedings of the Society for Information Display, v. 21/3, pp. 229-246.

CCIR Recommendation 500-3 (1986), Methods for the subjective assessment of the quality of television pictures, CCIR XVth Plenary Assembly, Dubrovnik, v.XI-1, pp. 165-173.

CCIR Recommendation 567-2 (1986), Transmission performance of television circuits designed for use in international connections, International Radio Consultative Committee's Recommendations and Reports of the CCIR, Transmission of Sound Broadcasting and Television Signals Over Long Distances (CMTT), v. XII.

CCIR Recommendation 654 (1986), Subjective quality of television pictures in relation to the main impairments of the analogue composite television signal, International Radio Consultative Committee's Recommendations and Reports of the CCIR, Broadcasting Service (Television), v. XI-1, pp. 223-230.

CCIR Report 313-6 (1986), Assessment of the quality of television pictures, International Radio Consultative Committee's Recommendations and Reports of the CCIR, Broadcasting Service (Television), v. XI-1, pp. 220-223.

CCIR Report 405-5 (1986), Subjective assessment of the quality of television pictures, International Radio Consultative Committee's Recommendations and Reports of the CCIR, Broadcasting Service (Television), v. XI-1, pp. 174-201.

CIE Supplement No. 2 to CIE Publication No. 15 (E-1.3.1) 1971/(TC-1.3.) (1978), Recommendations on uniform color spaces - color difference equations psychometric color terms.

Crow, E.L., Davis, F.A., and M. W. Maxfield (1960), Statistics Manual, Dover Publications Inc., 180 Varick Street, New York, New York.

Fink, D. G. (1975), Electronics Engineers' Handbook, McGraw-Hill Inc., pp. 20-3 to 20-22.

Geuen, W., and H. G. Preuth (1982), New performance criteria of edge detection algorithms, Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, v. 3, pp. 1936-1939.

Gonzalez, R. C., and P. Wintz (1987), Digital Image Processing, second edition, Addison-Wesley Publishing Company.

Held, R., Leibowitz, H. W., and H. L. Teuber (1978), Perception, Springer-Verlag Berlin Heidelberg New York, pp. 523-548.

Higgins, G. C. (1977), Image quality criteria, Journal of Applied Photographic Engineering, v. 3, n. 2, pp. 53-60.

Jain, A. K. (1989), Fundamentals of Digital Image Processing, Prentice-Hall Inc., Englewood Cliffs, New Jersey.

Limb, J. O. (1979), Distortion criteria of the human viewer, IEEE Transactions on Systems, Man, and, Cybernetics, v. 9, n. 12, December, pp. 778-793.

Mannos, J. L., and D. J. Sakrison (1974), The effects of a visual fidelity criterion on the encoding of images, IEEE Transactions on Information Theory, v. 20, n. 4, pp. 525-536.

Miyahara, M. (1988), Quality assessments for visual service, IEEE Communications Magazine, October, pp. 51-60.

Meiseles, H. (1988), Objective measurement methods of motion artifacts for 45 Mbit, NTSC, DPCM, bit-reduction video codecs, 130th SMPTE Technical Conference, October 15-19.

Miyahara, M., and Y. Yoshida (1988), Mathematical transform of (R, G, B) color data to Munsell (H, V, C) color data, SPIE Visual Communications and Image Processing, v. 1001, pp. 650-657.

Murakami, H., Hashimoto, H., and Y. Hatori (1988), Quality of band-compressed TV services, IEEE Communications Magazine, October, pp. 61-69.

Nesenbergs, M. (1989), Image data compression overview: issues and partial solutions, U.S. Department of Commerce, National Telecommunications and Information Administration, NTIA Report 89-252, October.

Newhall, S. M., Nickerson, D., and D. B. Judd (1943), Final report of the O.S.A. Subcommittee on the spacing of the Munsell colors, Journal of the Optical Society of America, v. 33, n. 7, pp. 385-418.

Ohtsuka, S., Inoue, M., and K. Watanabe (1988), Quality evaluation of pictures with multiple impairments based on visually weighted error, Society for Information Display International Symposium Digest of Technical Papers, v. XIX, May 24-26, pp. 428-431.

Owens, R., Venkatesh, S., and J. Ross (1989), Edge detection is a projection, Pattern Recognition Letters, v. 9, pp. 233-244.

Oppenheim, A. V., and R. W. Schafer (1975), Digital Signal Processing, Prentice-Hall, Englewood Cliffs, New Jersey.

Pearson, D. E. (1980), A three-stage process for the evaluation of image quality, Proceedings of the Society for Information Display, v. 21/3, pp. 271-278.

Sakrison, D. J. (1977), On the role of the observer and a distortion measure in image transmission, IEEE Transactions on Communications, v. 25, n. 11, November, pp. 1251-1267.

Shapley, R. M., and D. J. Tolhurst (1973), Edge detectors in human vision, J. Physiol., v. 229, pp. 165-183.

Taylor, J. M., Murch, G. M., and P. A. McManus (1989), TekHVC™: A uniform perceptual color system for display users, Proceedings of the Society for Information Display, v. 30/1, pp. 15-21.

Task, H. L. (1979), An evaluation and comparison of several measures of image quality, Aerospace Medical Research Lab Wright-Patterson AFB OH, AMRL-TR-79-7, January (available on Micro-fische, AD-A069 690).

Task, H. L., Pinkus, A. R., and J. P. Hornseth (1978), A comparison of several television display image quality measures, Proceeding of the Society for Information Display, v. 19/3, pp. 113-119.

Toit, T. C. du, and J. G. Lourens (1988), Automatic detection of image impairments, IEEE International Conference on Acoustics, Speech, and Signal Processing, Multidimensional Signal Processing, v. 2, April 11-14, pp. 1080-1083.

Tzafestas, S. G. (1986), Multidimensional Systems: Techniques and Applications, Marcel Dekker Inc, New York and Basel.

Tomich, D. J., Quincy, E. A., and D. Parsavand, (1989), Expert pattern recognition assessment of image quality, IEEE International Conference on Acoustics, Speech, and Signal Processing, Multidimensional Signal Processing, Audio & Electroacoustics, v. 3, May 23-26, pp. 1799-1802.

Westernik, J. H. D. M., and J. A. J. Roufs (1988), A local basis for perceptually relevant resolution measures, Society for Information Display International Symposium Digest of Technical Papers, v. XIX, May 24-26, pp. 360-363.

6. BIBLIOGRAPHY

Allnatt, J. W. (1983), Transmitted-Picture Assessment, John Wiley & Sons, New York.

Beaton, R. J. (1988), Linear systems metrics of image quality for flat-panel displays, SPIE Image Processing, Analysis, Measurement, and Quality, v. 901, pp. 144-151.

Goodman, J. S., and D. E. Pearson (1979), Multidimensional scaling of multiply-impaired television pictures, IEEE Transactions on Systems, Man, and Cybernetics, v. 9, n. 6, June, pp. 353-356.

Lewis, N. W., and J. W. Allnatt (1965), Subjective quality of television pictures with multiple impairments, Electronic Letters, v. 1, n. 7, September, pp. 187-188.

Lourens, J. G., and M. W. Coetzer (1987), Image impairment detection using digital signal processing, International Conference of Television Measurements, 3rd, pp. 33-38.

Moon, D. L. (1988), Image quality for discrete-element displays: variables, metrics, and measurements, SPIE Image Processing, Analysis, Measurement, and Quality, v. 901, pp. 161-170.

Netravali, A. N., and B. G. Haskell (1988), Digital Pictures Representation and Compression, Plenum Publishing Corporation, New York, New York, pp. 245-297.

Pratt, W. K. (1978), Digital Image Processing, John Wiley & Sons, New York, pp. 162-200.

Smith-Kerker, P. L., and R. G. Bias (1988), The effect of color encoding on the subjective quality of color images, Society for Information Display International Symposium Digest of Technical Papers, v. XIX, May 24-26, pp. 85-88.

7. APPENDIX A: EQUATIONS

This appendix describes all of the equations that were used to compute the feature values in this report.

Equation (1): Single-frame Temporal Alignment

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 $v = \{1, 2, ..., N_v\}$, and $h = \{1, 2, ..., N_h\}$, 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 video sequence and the output video sequence are sampled at the same frame rate. Then, given an output reference frame $t_o = r$, the single-frame temporal alignment problem is to find the closest corresponding input frame $t_i = m$. The single-frame 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 standard deviation of the error (accumulated over all pixels in the error image) or

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

where t_i falls within the range from t_l to t_u, inclusive. Pairing of the input and output video frame indices is performed as ..., ($t_i = m-1$, $t_o = r-1$), ($t_i = m$, $t_o = r$), ($t_i = m+1$, $t_o = r+1$), ...

Equation (2): Missing Frame Ratio (MFR)

To apply multi-frame temporal alignment to the output video frame sequence $o(v,h,t_o)$, where $t_o = \{1, 2, ..., N_o\}$, and N_o is the total number

of output video frames, the method of finding the closest matching input video frame (equation 1, Appendix A) is applied to every output video frame $t_o = \{1, 2, ..., N_o\}$. 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 multi-frame alignment, the closest input video frame index to each one of the output video frames is stored. Let the number of <u>unique</u> 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

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

<u>Equation (3): Mean of the Sobel Image (M-SI)</u>

If the Sobel edge extracted image is given by s(v,h), where v and h represent the vertical and horizontal sampling indices of the video image, then the mean of s(v,h) over a given sub-regional area is given by

$$M\text{-}SI = \frac{1}{N_A}\sum_{v}\sum_{h} s(v,h)$$

where the summation is performed over the sub-regional area and N_A is the total number of pixels within the sub-regional area.

<u>Equation (4): Standard Deviation of the SI (SD-SI)</u>

Following the notation established for equation (3) in Appendix A, the standard deviation of the sub-regional Sobel image is

$$SD\text{-}SI = \sqrt{\{\frac{1}{N_A}\sum_v \sum_h s(v,h)^2\} - (M\text{-}SI)^2}$$

Equation (5): Root Mean Square of the SI (RMS-SI)

Following the notation established for equation (3) in Appendix A, the root mean square of the sub-regional Sobel image is

$$RMS\text{-}SI = \sqrt{\frac{1}{N_A}\sum_v \sum_h s(v,h)^2}$$

Equation (6): Number of Pixels Greater than Threshold of SI (NPGT-SI)

Following the notation established in equation (3) in Appendix A and letting T be the chosen threshold, the number of pixels greater than T within the sub-regional Sobel image is

$$NPGT\text{-}SI = \sum_v \sum_h u(v,h)$$

$$\textit{where}$$

$$u(v,h) = 1 \ \textit{if} \ s(v,h) > T$$
$$= 0 \ \textit{otherwise}$$

Equation (7): Mean of the Positive Sobel Difference Image (M-PSDI)

If the Sobel difference image (Sobel filtered input image minus the Sobel filtered output image) is given by $s_d(v,h)$, where v and h represent the vertical and horizontal sampling indices, then the mean of the sub-regional, positive part of the sobel difference image is

$$M\text{-}PSDI = \frac{1}{N_A}\sum_v \sum_h s_d(v,h) \quad , \; s_d(v,h)>0$$

where N_A is the total number of points within the sub-regional area.

Equation (8): Standard Deviation of the PSDI (SD-PSDI)

Following the notation of equation (7) in Appendix A, the standard deviation of the sub-regional, positive part of the Sobel difference image is

$$SD\text{-}PSDI = \sqrt{\frac{1}{N_A}\sum_v \sum_h s_d^2(v,h) - (M\text{-}PSDI)^2} \quad , \; s_d(v,h)>0$$

Equation (9): Root Mean Square of the PSDI (RMS-PSDI)

Following the notation of equation (7) in Appendix A, the root mean square of the sub-regional, positive part of the Sobel difference image is

$$RMS\text{-}PSDI = \sqrt{\frac{1}{N_A}\sum_v \sum_h s_d^2(v,h)} \quad , \; s_d(v,h)>0$$

Equation (10): Number of Pixels Greater than Threshold of PSDI (NPGT-PSDI)

Following the notation of equation (7) in Appendix A, and letting T_p be the chosen threshold, the number of pixels greater than T_p within the sub-regional, positive part of the Sobel difference image is

$$NPGT\text{-}PSDI = \sum_v \sum_h u_p(v,h)$$

where

$$u_p(v,h) = 1 \ \ if \ s_d(v,h) > T_p$$
$$= 0 \ \ otherwise$$

Equation (11): Mean of the Negative Sobel Difference Image (M-NSDI)

If the Sobel difference image (Sobel filtered input image minus the Sobel filtered output image) is given by $s_d(v,h)$, where v and h represent the vertical and horizontal sampling indices, then the mean of the sub-regional, negative part of the sobel difference image is

$$M\text{-}NSDI = \frac{1}{N_A} \sum_v \sum_h s_d(v,h) \quad , \ s_d(v,h) < 0$$

where N_A is the total number of points within the sub-regional area.

Equation (12): Standard Deviation of the NSDI (SD-NSDI)

Following the notation of equation (11) in Appendix A, the standard deviation of the sub-regional, negative part of the Sobel difference image is

$$SD\text{-}NSDI = \sqrt{\frac{1}{N_A} \sum_v \sum_h s_d^2(v,h) - (M\text{-}NSDI)^2} \quad , \ s_d(v,h) < 0$$

Equation (13): Root Mean Square of the NSDI (RMS-NSDI)

Following the notation of equation (11) in Appendix A, the root mean square of the sub-regional, negative part of the Sobel difference image is

$$RMS\text{-}NSDI = \sqrt{\frac{1}{N_A}\sum_v \sum_h s_d^2(v,h)} \quad , \; s_d(v,h)<0$$

Equation (14): Number of Pixels Less than Threshold of NSDI (NPLT-NSDI)

Following the notation of equation (11) in Appendix A, and letting T_n be the chosen threshold, the number of pixels less than T_n within the sub-regional, negative part of the Sobel difference image is

$$NPLT\text{-}NSDI = \sum_v \sum_h u_n(v,h)$$

$$where$$

$$\begin{aligned} u_n(v,h) &= 1 \;\; if \;\; s_d(v,h)<T_n \\ &= 0 \;\; otherwise \end{aligned}$$

Equation (15): Temporal Root Mean Square Position Error (TRMS-PE)

Let the input, or reference, vertical and horizontal positions of the moving object be represented by $v_i(t_i)$ and $h_i(t_i)$, where t_i represents the frame sampling index such that $t_i = \{1, 2, 3, ..., N_i\}$, and N_i is the total number of time samples for the input object path. Similarly, let the output vertical and horizontal positions of the moving object be represented by $v_o(t_o)$ and $h_o(t_o)$, where $t_o = \{1, 2, 3, ..., N_o\}$, and N_o is the total number of time samples for the output object path. In order to measure TRMS-PE, the input and output motion paths have to be aligned to compensate for the absolute video delay of the device under test. The alignment procedure described here corresponds to what a viewer would observe if that viewer was insensitive to the absolute video delay. Assume that the output motion path corresponds to some portion of the input path, and is thus contained completely within the input motion path (i.e., $N_o < N_i$). Then, the TRMS-PE feature is computed as the minimum root mean square position error of the output motion path with respect to the input motion path, where the minimization is performed over all possible time shifts $s = \{0, 1, 2, ..., N_i\text{-}N_o\}$ of the two motion paths. In equation form, the computation may be written as

$$TRMS\text{-}PE = \underset{s}{MIN} \sqrt{\frac{1}{N_o} \sum_{t_o=1}^{t_o=N_o} [v_i(t_o+s) - v_o(t_o)]^2 + [h_i(t_o+s) - h_o(t_o)]^2}$$

As in equation (1) in Appendix A, a priori knowledge of the absolute video delay may be used to narrow the range of time shifts.

Equation (16): Standard Deviation of Difference Image (SD-DI)

Assume that the input and output video sequences have been aligned using single frame temporal alignment given in equation (1) in Appendix A, and thus each output video frame has been paired with some input video frame. Let each pair of video frames be represented by the index p = {1, 2, 3, ..., N_p), where N_p is the total number of input/output pairs. Let the difference image (input image minus output image) of each input/output pair be represented by $d_p(v,h)$, where v and h are the vertical and horizontal sampling indices. Then, the standard deviation of the difference image over a sub-regional area is given by

$$(SD\text{-}DI)_p = \sqrt{\{\frac{1}{N_A}\sum_v \sum_h d_p^2(v,h)\} - \{\frac{1}{N_A}\sum_v \sum_h d_p(v,h)\}^2}$$

where N_A is the total number of points within the sub-regional area.

Equation (17): Temporal Mean of SD-DI (TM-SD-DI)

Following the notation of equation (16) in Appendix A, the temporal mean of the time history of SD-DI is given as

$$TM\text{-}SD\text{-}DI = \frac{1}{N_p} \sum_{p=1}^{p=N_p} (SD\text{-}DI)_p$$

Equation (18): Temporal Standard Deviation of SD-DI (TSD-SD-DI)

Following the notation of equation (16) and (17) in Appendix A, the temporal standard deviation of the time history of SD-DI is given as

$$TSD\text{-}SD\text{-}DI = \sqrt{\left\{ \frac{1}{N_p} \sum_{p=1}^{p=N_p} (SD\text{-}DI)_p^{\,2} \right\} - (TM\text{-}SD\text{-}DI)^2}$$

Equation (19): Temporal Root Mean Square of SD-DI (TRMS-SD-DI)

Following the notation of equation (16) in Appendix A, the temporal root mean square of the time history of SD-DI is given as

$$TRMS\text{-}SD\text{-}DI = \sqrt{\frac{1}{N_p} \sum_{p=1}^{p=N_p} (SD\text{-}DI)_p^{\,2}}$$

8. APPENDIX B: FILTERS

This appendix describes the median filter that was used to precondition the digitized video images and the Sobel filter that was used to extract edges from the preconditioned images.

Median Filter

A median filter will remove noise spikes from the image without significantly blurring the edges. Very fine detail, such as sharp corners, will be removed by the median filter. For this report, a 3 x 3 median filter was used. Figure B-1 shows 9 image pixel values (X_1, X_2, ..., X_9) within the 3 x 3 filter window. In Figure B-1, the 3 x 3 filter window is centered on the image pixel value X_5.

X_1	X_2	X_3
X_4	X_5	X_6
X_7	X_8	X_9

Figure B-1. Filter window centered on image pixel value X_5.

The median filter outputs the image pixel value that is the median of the 9 image pixel values (X_1, X_2, ..., X_9). That is, the 9 pixel values are first sorted from low to high, and then the middle value is selected as the median. The median filtered image is obtained by sliding the 3 x 3 window over the entire input image. At each pixel for which the mask is centered in the input image, the median value is placed in the output image. Note that as an edge is crossed, one side or the other dominates the window and the output switches sharply.

Sobel Filter

The Sobel filter is an edge extraction filter that is implemented using two filters. One filter is designed to extract horizontal edges from the image and the other filter is designed to extract vertical edges. The outputs from the two filtering operations are then combined to give a composite edge extracted image. Figure B-2 gives the filter

mask that extracts the horizontal edges. Figure B-3 gives the filter mask that extracts the vertical edges.

-1	-2	-1
0	0	0
1	2	1

Figure B-2. Horizontal edge extraction filter mask.

-1	0	1
-2	0	2
-1	0	1

Figure B-3. Vertical edge extraction filter mask.

If both of the masks shown in Figures B-2 and B-3 are centered on pixel value X_5, as in Figure B-1, then the output response at pixel location X_5 from the horizontal edge extraction filter is

$$G_h = -1*X_1 - 2*X_2 - 1*X_3 + 1*X_7 + 2*X_8 + 1*X_9$$

and the output response at pixel location X_5 from the vertical edge extraction filter is

$$G_v = -1*X_1 + 1*X_3 - 2*X_4 + 2*X_6 - 1*X_7 + 1*X_9$$

Note that horizontal edges result in an output response G_h and vertical edges result in an output response G_v. Diagonal edges result in output responses G_h and G_v. The composite output response at pixel location X_5 from both filters is computed as

$$G = [G_h^2 + G_v^2]^{1/2}$$

The output image pixel values are obtained by computing the filter response G at each corresponding pixel in the input image, where both filter masks (Figure B-2 and B-3) are centered on the input image pixel.