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

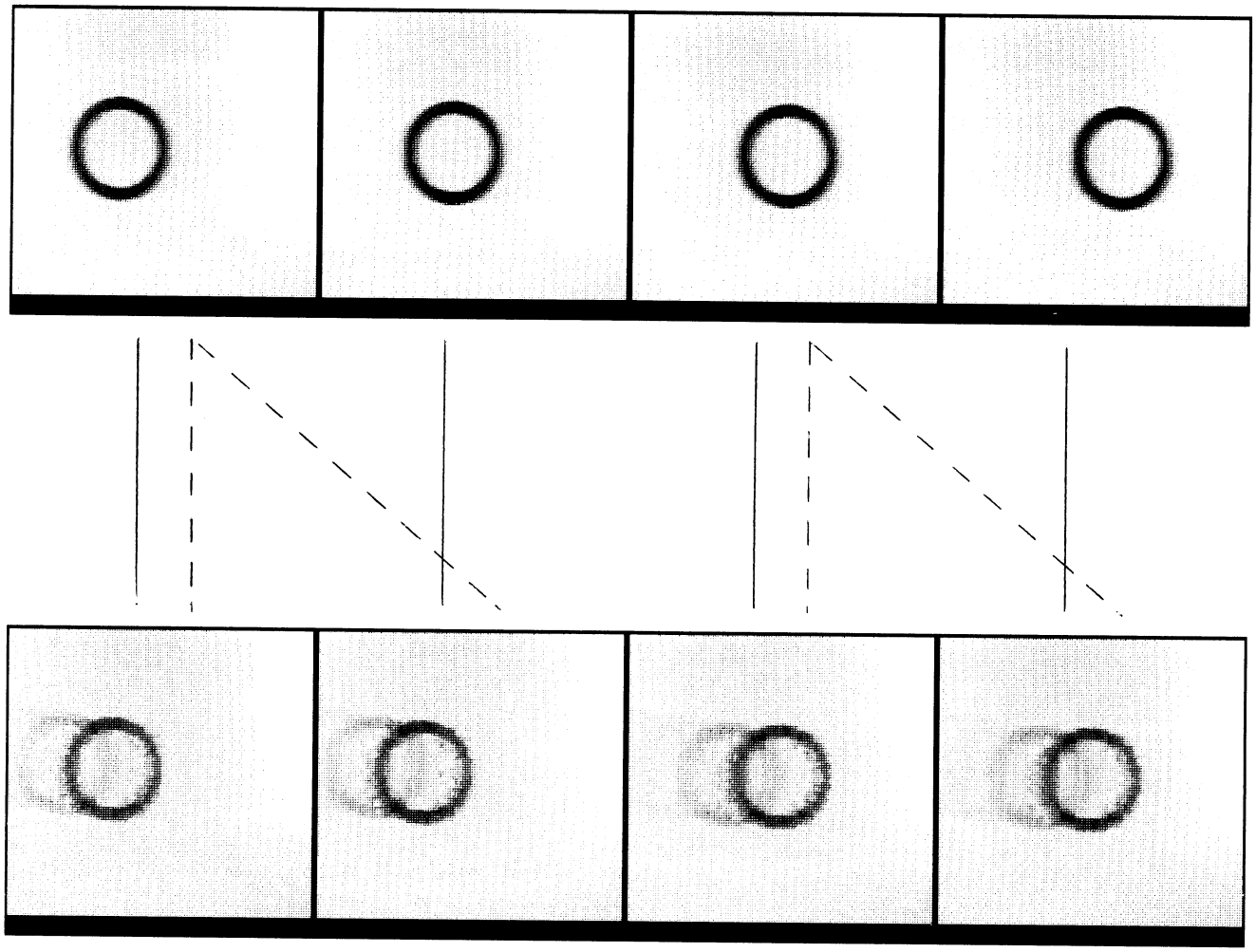

Figure 2. Single-frame and multi-frame alignment. Top row - original NTSC grabbed every 1/30 sec from left to right. Bottom row - VTC/VT codec output. Solid lines represent single-frame alignment and dashed lines represent multi-frame alignment.

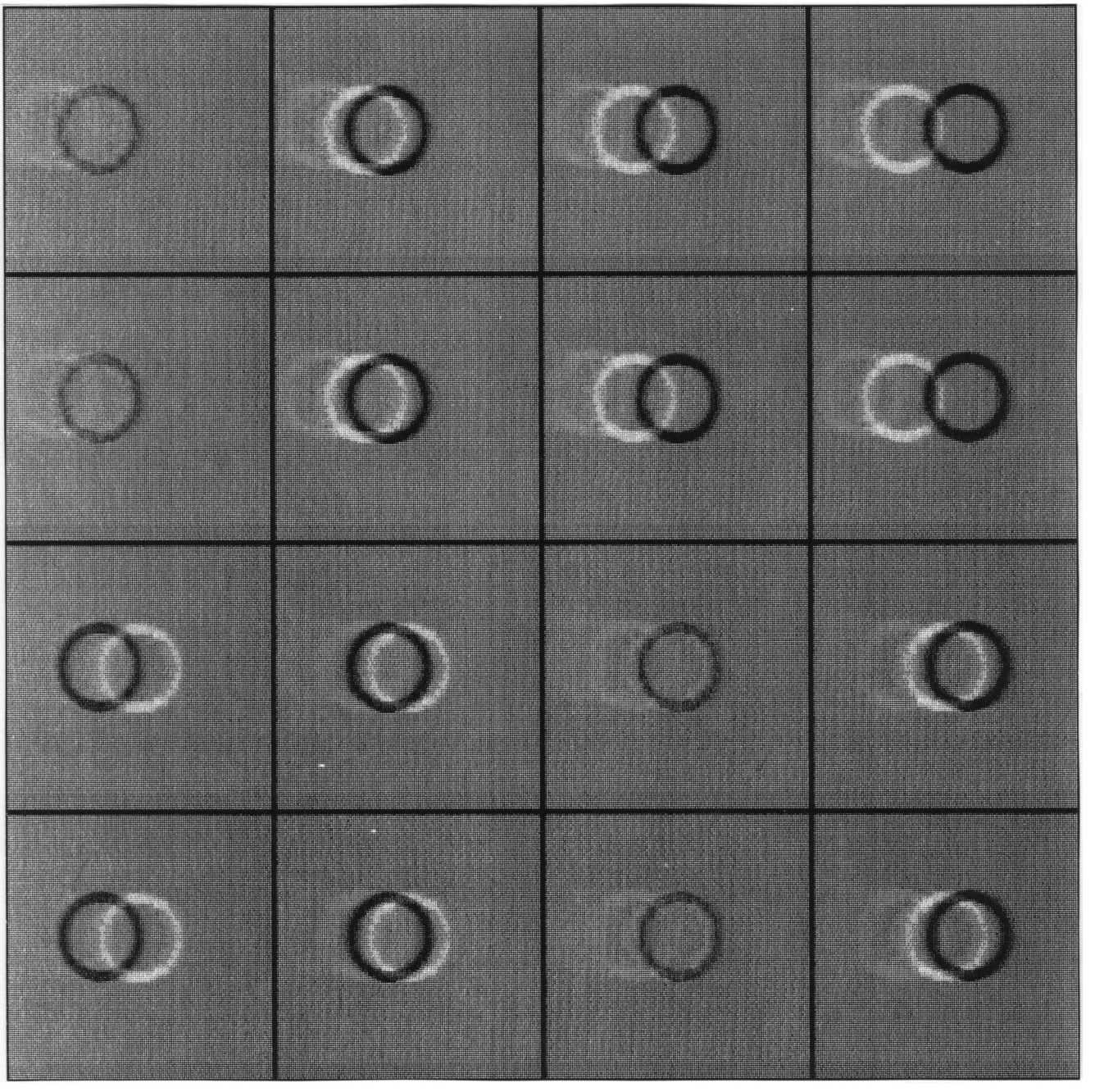

Figure 3. Error difference images (input–output) of Figure 2. Top row – NTSC input (top row in Figure 2) minus codec output image 1 (bottom row, leftmost frame in Figure 2). Second, third, and fourth rows are NTSC input minus codec output images 2, 3, and 4, respectively.

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.