COMMITTEE T1
CONTRIBUTION

Document Number: T1A1.5/92-139

****

STANDARDS PROJECT:	Analog Interface Performance Specifications for Digital Video Teleconferencing/Video Telephony Service
TITLE:	Preliminary Results of One-Way Video Delay Measurement Algorithms
ISSUE ADDRESSED:	One-Way Video Delay of VTC/VT
SOURCE:	National Telecommunications and Information Administration Institute for Telecommunication Sciences (Coleen Jones, Stephen Wolf, and Margaret Pinson)
DATE:	July 13, 1992
DISTRIBUTION TO:	T1A1.5
KEYWORDS:	Video Quality, Video Performance Specifications, Objective Quality, Subjective Quality
DISCLAIMER:	

****

1. Introduction

The Institute for Telecommunication Sciences (ITS) has been investigating the measurement of one-way video delay through digital transport systems with the aim of assisting the T1A1.5 VTC/VT Sub-working group in the development of the VTC/VT ANSI standard. Delay exists for any system; it is relevant to VTC systems in that it is known to interfere with interactive human communication. Also, a valid measurement of video delay is the first step toward measurement of audio-visual interactions such as lip synchronization.

Out-of-service measurements can be used to obtain a baseline system delay. However, note that in many systems, video delay, as with video performance, is dependent upon the scene content. In-service measurements are therefore necessary to measure the dynamic nature of delay through digital systems. This contribution proposes a method for dynamically measuring the one-way video delay of a transmission service channel using information that has previously been shown to be good for measuring the temporal distortion of a video signal. The following discussion first describes the problem of video alignment and the goals of our alignment algorithms. This is followed by a description of our two algorithms and the results to date. This contribution then concludes with a block diagram of a potential, real-time implementation.

Given the contents of this contribution, we recommend that a new performance parameter called “Overall Path Delay” be added to the in-service baseband VTC/VT Performance Specification, section 6.3 of the draft VTC/VT Standard (document T1A1.5/92-107). We also recommend that the techniques presented here be considered for measuring this new in-service performance parameter, as well as the corresponding out-of-service performance parameter given in section 5.3.1 of the draft VTC/VT Standard.

2. Description of Problem

The measurement of video delay and time alignment of the original and degraded video are inherently the same problem. Measuring one results in knowing the other. Time alignment of digitally transported signals is a difficult task because there is not a one-to-one correspondence between input and output frames. Two example degradations seen in digital coding that affect the correspondence between original and degraded frames are frame repetition and interframe prediction. In the former, input frames are lost, and in the later, output frames are created that do not correspond to any single input frame. This brings up the question of what is correct alignment? The original and degraded video can be compared frame-by-frame by a human viewer, but this is highly tedious and prone to error. One solution to this dilemma is to use an objective method for aligning video, the results of which can be verified visually. The visual results can then support or disprove the given alignment method.

With these thoughts in mind, what are some likely methods that could be used for video alignment? One obvious method is using the minimum mean square error between an original video frame and a degraded video frame for each frame in the scene. The two images yielding the minimum error can be considered aligned. An example of this method can be seen in Figure 1 which plots the original versus

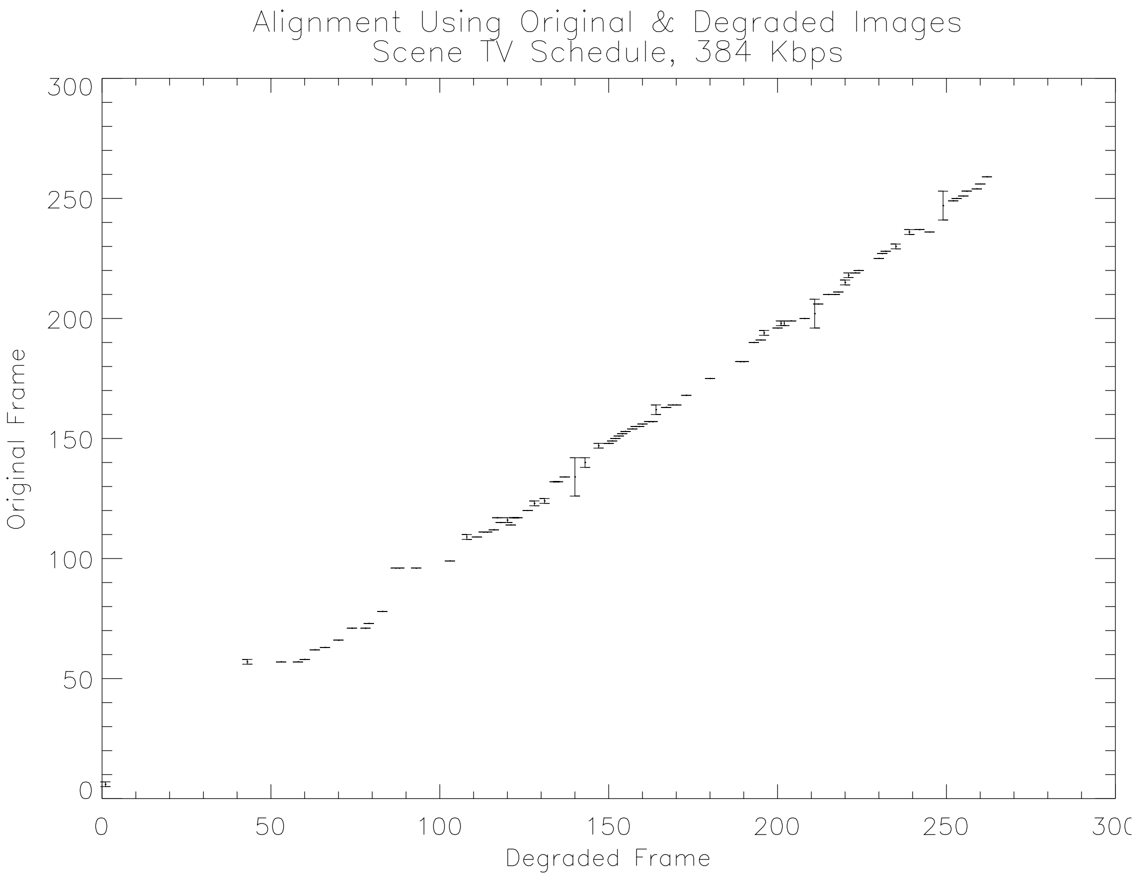


Figure 1 Alignment results using the minimum mean square error between original and degraded images.

degraded frame for a graphics scene (a TV schedule). There is not a data point for each degraded frame because for some degraded frames, there was not a unique original frame found. Alignment using minimum mean square error can be uncertain if there is not a well defined minimum value. Any frame surrounding the absolute minimum which is within the system error may also be a valid alignment frame. This uncertainty is represented by the error bars. From this figure, it is evident that the delay does indeed vary. Also, this method is computationally intensive and requires that both the original and degraded images be available at one location, which makes this method impractical for many applications.

This contribution discusses an approach where the information required for correct alignment is simply a scalar value per frame derived from the original and degraded images. We have been investigating the standard deviation of the temporal first order difference image as a parameter for use in such an alignment algorithm. The standard deviation reduces each frame to a scalar number, and the time history can be written as a vector. The alignment problem then becomes one of aligning the vector derived from the original video sequence, $\text{std}_{\text{space}}(\boldsymbol{O}_n - \boldsymbol{O}_{n-1})$, with the vector derived from the degraded video sequence, $\text{std}_{\text{space}}(\boldsymbol{D}_n - \boldsymbol{D}_{n-1})$. An example plot of these two vectors can be seen in Figure 2. This is the same data used as an example

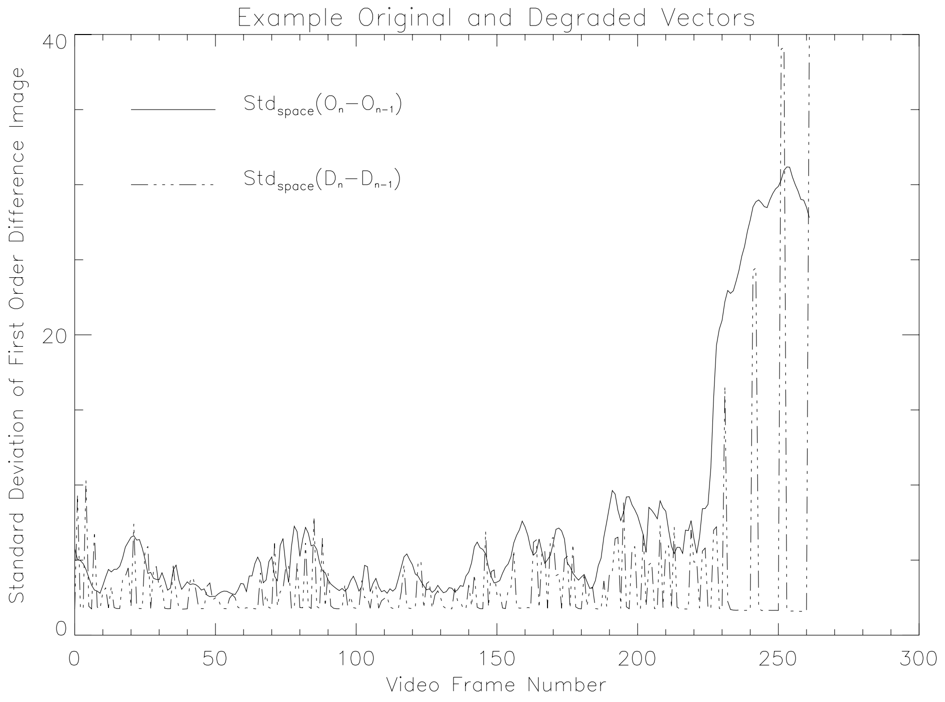
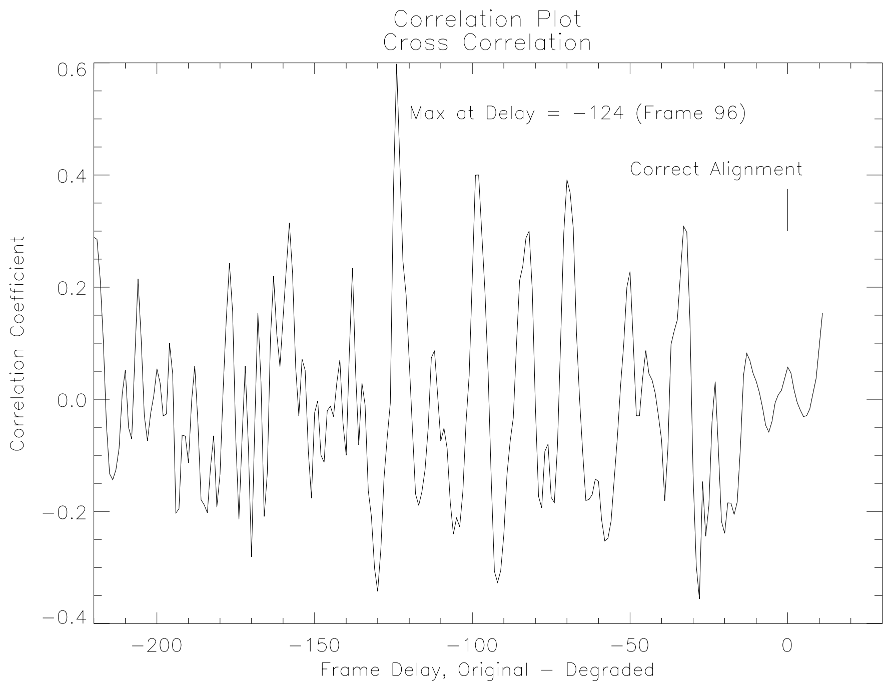


Figure 2 Example original and degraded vectors.

in contribution T1A1.5/92-138 contributed by NTIA/ITS. The original vector is a smoothly varying curve, where the large amplitudes represent a pan. The degraded vector is spiky however. The low amplitudes are frame repetitions, and the large amplitude spikes are frame updates. These vectors are used in our quality prediction algorithm, and thus do not represent an increase in data sent from input to output in a real-time system.

Representing the time history of video frames with a vector of scalars allows other correlation measures such as cross correlation to be investigated. Cross correlation works best with Gaussian distributed data, but our data is non-Gaussian. Cross correlation therefore works poorly because it aligns the shapes of the waveforms independent of their amplitudes. Since our data contains waveforms from codecs which use frame repetition, the amplitude of the degraded waveform relative to the original waveform is significant. Some degraded amplitudes are similar to the original amplitudes and some are similar to the noise floor. Figure 3 (top) is an example cross correlation plot. The abscissa is the delay between the shifted original alignment vector element and the fixed degraded alignment vector element (degraded element 220). The correlation coefficient has no distinguishable peak because the two waveforms have similar shapes at different times. The maximum value occurs at a delay of -124 (original element 96), whereas the correct alignment



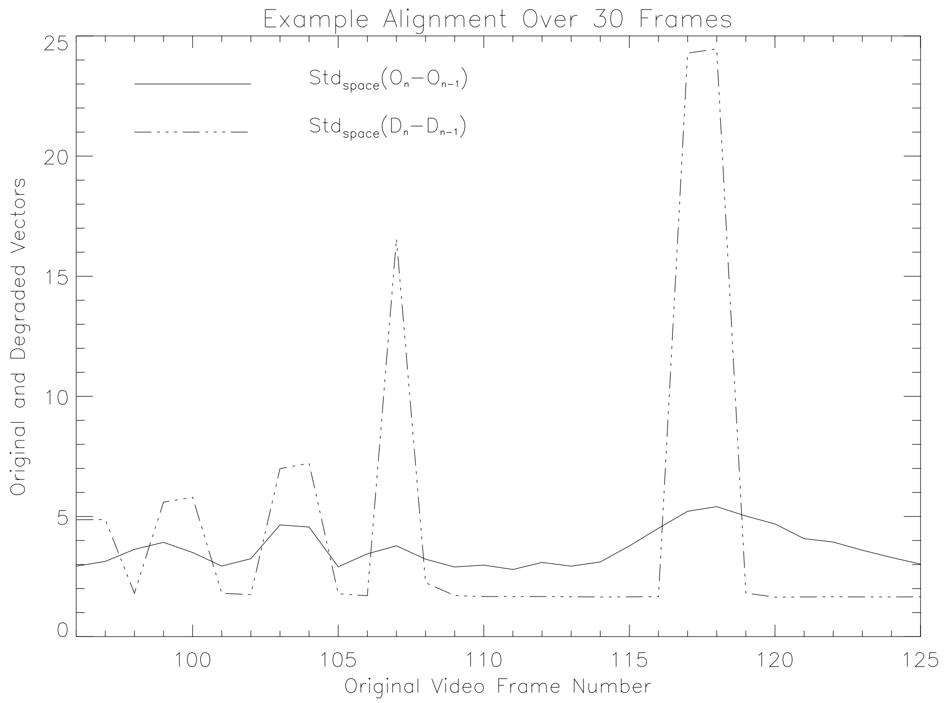


Figure 3 Example correlation plot using cross correlation (top). Alignment results for above correlation plot (bottom).

occurs at a delay of zero (degraded element 220). Figure 3 (bottom) shows the window of the original alignment vector, X, which was chosen as the correct alignment for the window of the degraded alignment vector, Y. The cross correlation algorithm aligned elements 220-250 of the degraded alignment vector with elements 96-126 of the original alignment vector. This is a good example of cross correlation aligning the shape of the waveform independent of amplitude.

The next section discusses our two alignment algorithms. The first method discussed is the alignment by histogram method, which is a statistical method. The second is alignment by peaks, which is an empirical method.

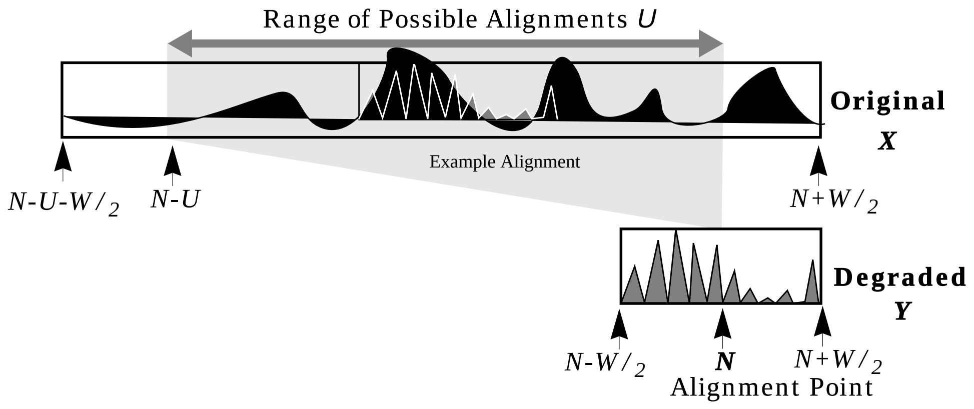
3. Description of Algorithms

Video is an inherently discrete signal having approximately 30 frames per second. Our algorithms are therefore written using discrete notation. When a variable is subscripted, such as $\boldsymbol{O}_n$, we are referring to the nth frame, or a scalar value derived from the nth frame within a time series of frames. Note that video systems are causal. There can be no output prior to an input. The following is a list of symbols used in describing the algorithms:

n: variable time index (frame)
N: constant (fixed) time index (frame)
k: shift index
$\boldsymbol{O}_n$: nth original video frame
$\boldsymbol{D}_n$: nth degraded video frame
X: original alignment vector
Y: degraded alignment vector
$X_n = \mathrm{std}_{\mathrm{space}}(\boldsymbol{O}_n - \boldsymbol{O}_{n-1})$: nth element of original alignment vector
$Y_n = \mathrm{std}_{\mathrm{space}}(\boldsymbol{D}_n - \boldsymbol{D}_{n-1})$: nth element of degraded alignment vector
$\tilde{X}_n$: quantized value of X_n
$\tilde{Y}_n$: quantized value of Y_n
C_k: correlation value associated with a shift of k
W: time window width over which the correlation is performed
U: uncertainty
m: mean value
$\forall$: symbol meaning “for all“

Both alignment algorithms generate a correlation vector by sliding a fixed window on the degraded alignment vector across a shifting window on the original alignment vector. This is demonstrated in Figure 4. One element of the correlation vector is calculated for each shift across U, the uncertainty.

The time alignment uncertainty U, is the region within which the correctly aligned original alignment vector element X_n can be found. It will never exceed the



W	Window width of the degraded alignment vector to be aligned.
$Y_{N-W/2}$	Start of the degraded window; no previous degraded alignment vector elements are required.
Y_N	Current degraded alignment vector element to be aligned at the center of window W.
$Y_{N+W/2}$	End of the degraded window; no later degraded alignment vector elements are required.
U	Uncertainty across original alignment vector.
$X_{N-U-W/2}$	The first original alignment vector element required.
X_{N-U}	The earliest original alignment vector element considered for alignment.
X_N	The last original alignment vector element considered for alignment.
$X_{N+W/2}$	The last original alignment vector element required.
n	Free time variable.

Figure 4 Configuration of original and degraded vectors.

maximum system delay. The time window width W, is the region in the degraded alignment vector which contains the desired degraded alignment vector element Y_N. A smaller window allows us to track small variations in delay, but this is at the expense of correct alignment in some cases. A larger window increases the accuracy of the alignment, but decreases the sensitivity to variations in delay; it is also more computationally expensive. It is desirable to track this variable delay, because the video delay of a transmission service channel depends upon the motion in the video scene. The selected window width therefore involves a trade off between accuracy and sensitivity.

As discussed earlier, the original and degraded values X_n and Y_n, will be used for alignment as opposed to using the original and degraded images. In a real-time system, this would make it possible to transmit data from the input to the output where the alignment computation can be made. Figure 5 shows a block diagram of such a system. The original video is sampled by a PC, where the scalar values are calculated and time-tagged using a satellite clock. The scalar values are then transmitted to a PC located at the decoder output. This PC calculates the same scalars from the sampled degraded video. The original and degraded scalar values

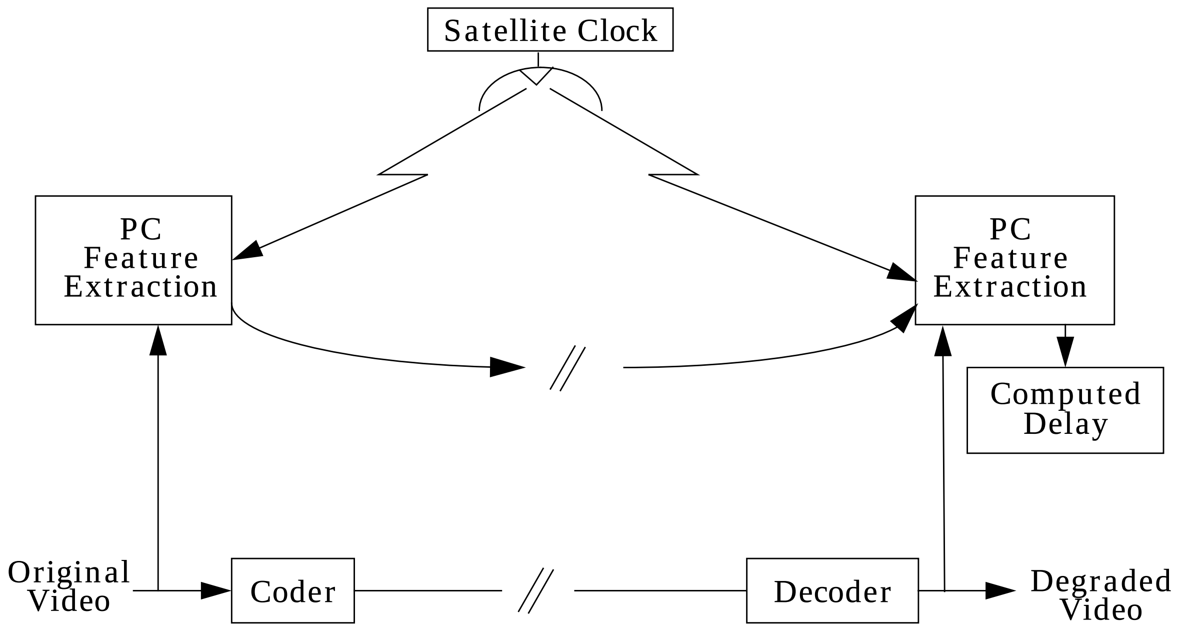


Figure 5 Block Diagram - Objective Delay Measurement.

can then be used to calculate the objective quality and the one-way video delay using an alignment algorithm and the satellite time stamps.

3.1 Correlation by Histogram Table

The first alignment method we investigated is referred to as Maximum Product of Conditional Probabilities Alignment[1]. This method is based upon the statistics of correctly aligned original and degraded data. For example, if the conditional probability of the degraded data given the original data is Gaussian distributed about the original data with variance σ^2, then

$$p(Y_n / X_{n-k}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(Y_n - X_{n-k})^2}{2\sigma^2}\right). \tag{1}$$

Assuming sample independence, the likelihood that a given alignment k is correct is given by the product $\prod_n p(Y_n / X_{n-k})$. It can be shown that under the assumption that if the time window width W, is significantly larger than the shift k, this problem reduces to the traditional multiplicative cross correlation algorithm.

If k is defined as the offset (shift) of the original alignment vector element with respect to the N^{th} degraded alignment vector element Y_N, the alignment correlation value at original time (frame) N-k is denoted C_k. The correlation value is the product of conditional probabilities as discussed above. If we take the logarithm of both sides

1. The idea of using conditional probabilities for solving the time delay estimation problem was first made known to one of the authors, Stephen Wolf, by Kevin Miller and Alden Park at the Naval Weapons Center, China Lake, CA in 1986.

of the product equation, the logarithm of the product becomes the sum of the logarithms of the conditional probabilities. Thus, the correlation can be written as

$$\log_{10}(C_k) = \sum_{j=-W/2}^{W/2} \log_{10} p\,(\tilde{Y}_{N+j}/\tilde{X}_{N-k+j}) \quad \text{where} \quad (0 \le k \le U)\,. \tag{2}$$

The delay is the shift k_0 which maximizes C_k. The original alignment vector element which aligns to the degraded alignment vector element Y_N is X_{N-k_0}.

If the statistics of the degraded and original alignment vectors are not known in a closed mathematical form *a priori*, then experimental conditional probabilities may be used. The larger and more representative the data set from which the experimental conditional probabilities are gathered, the better this method will perform. Given that our experimental data is not easily described in closed form, our conditional probabilities were generated experimentally. Our results were obtained by calculating the alignment correlation value using a static, histogram look-up table. The alignment correlation look-up table was computed as follows. Each histogram in the table is a conditional probability density function (pdf) $p\,(\tilde{Y}_n/\tilde{X}_{n-k})$ where $\tilde{X}_{n-k}$ is the quantized original element and $\tilde{Y}_n$ is the quantized degraded element. $p\,(\tilde{Y}_n/\tilde{X}_{n-k})$ then represents the probability that Y_n falls within some range, given that $X_{n\text{-}k}$ falls within some range. These probabilities are initialized by empirically computing their value for a set of vectors correctly aligned by some other method.

In general, both the original and degraded alignment vector elements may be quantized non-linearly, for example, using the method of Linde, Buzo, and Gray [1]. The original elements, being distributed approximately exponential, were quantized exponentially. The conditional probability density functions in the table were complex functions. Each histogram was quantized uniformly for ease of computation, but uniform quantization is not optimal for these conditional probability density functions. Uniform quantization allows direct calculation of both the original and degraded bin numbers. Direct calculation of the bin numbers results in significant computational savings for a real-time system.

The histogram algorithm operates as follows. This algorithm performs alignment at the current degraded alignment vector element, labeled Y_N. In a real-time system, Y_N would be incremented 30 times per second. Both the original X, and degraded Y, vectors are buffered for a given amount of time (See Figure 4). If the system has a known maximum delay or uncertainty of U, the correct alignment cannot occur before the $N\text{-}U^{th}$ original element. Hence, the original alignment vector is searched from $X_{N-U} \ldots X_N$ for the correct alignment for degraded alignment vector element Y_N. Original elements in this range are referred to by $X_{N\text{-}k}$, where k ranges from 0 up to U. Comparison between element Y_N and element $X_{N\text{-}k}$ is made by comparing a window of size W centered around Y_N with a window of size W centered around $X_{N\text{-}k}$. These two windows are $[Y_{N\text{-}W/2}, Y_{N+W/2}]$, and $[X_{N\text{-}k\text{-}W/2}, X_{N\text{-}k+W/2}]$, respectively.

The histogram alignment can therefore be generally described by the following steps:

1 Find C_k (the probability that degraded alignment vector element Y_N aligns with

original alignment vector element X_{N-k}) for each of the U possible alignments; k = 0 to U.

1.1 For each element n in window W, ($n = N\text{-}W/2$ *to* $N+W/2$), look up the probability of Y_n given X_{n-k} which is approximated by $p(\tilde{Y}_n/X_{n-k})$ in the histogram look-up table.

1.2 Take the logarithm of each of these W probabilities from step 1.1.

1.3 Sum the logarithms of the W probabilities from step 1.2, to calculate C_k using equation (2).

2 Choose shift k_0, such that C_k is maximized.

Conditional probabilities allow us to characterize alignment based upon observed amplitudes of the original and degraded alignment vectors X and Y. For example, for high bit-rate codecs and analog systems, we can say that given the original value, it is very likely that the degraded amplitude will be similar to that of the original. However, if we consider codecs which perform frame-repetition, we can say that given the original value, the degraded amplitude is likely to be either similar to the original or similar to the system noise level (dissimilar to the original). The larger and more representative the histogram training data, the more accurate these probabilities will become. Results using this method will be presented in section 4.

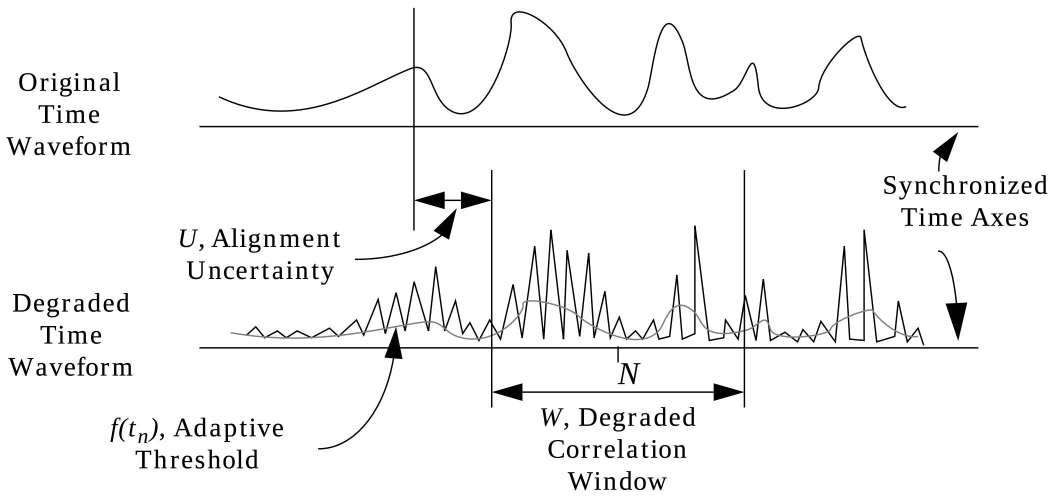
From a real-time system standpoint, this algorithm contains few redundancies. The largest savings come from applying the log function to the histograms as they are created. This simplifies the calculation of C_k by eliminating step 1.2. The quantization of X can be computed once and saved, the first time each element is needed, and thereafter the bin number can automatically be calculated. This gives a slight savings, because the time required to quantize each element X_n, every time it is used is greater than the lookup time required to access the array of quantized X values.

Neither of the above are enough to significantly affect the order of magnitude of this algorithm, which is $O(U \cdot W)$ operations. The uncertainty U, was 60 frames, and the window width W, was 120 frames. This means that the algorithm operates as a function of its slowest, largest inner loop (step 1.1) which is executed $U \cdot W$ times. The entire algorithm takes approximately 60 ms on a 50 MHz, 486 processor; which fails the real time requirement of 33 ms. One could, however, compute the alignment less frequently than every frame.

3.2 Correlation by Peaks

If the statistics of properly aligned original and degraded alignment vectors derived from the original and degraded images are not known *a priori,* then one must use an alignment algorithm that works well for a wide range of video systems. The following describes an alignment algorithm using only the peaks of the original and degraded alignment vectors. Thus, the points of the degraded alignment vector where the codec is performing frame repetition are not used for alignment.

Referring to Figure 6, the first step in the alignment algorithm is to compute a



Original Time Waveform
Synchronized Time Axes
U, Alignment Uncertainty
Degraded Time Waveform
N
$f(t_n)$, Adaptive Threshold
W, Degraded Correlation Window

Figure 6 Correlation by peaks alignment algorithm description.

dynamic threshold that will be used to separate peaks from valleys in the degraded alignment vector. The noise floor is dynamic depending upon the measurement system and the transmission service channel under test. A dynamic threshold will adapt to different noise floors, thus giving better performance when the peaks of the degraded alignment vector approach the noise floor. This condition is present when the video scene contains very small amounts of motion (e.g., only the lips and eyes are moving). The adaptive threshold is generated by filtering the degraded alignment vector with a low-pass Finite Impulse Response (FIR) filter. The dashed line in Figure 6, $f(t_n)$, illustrates the adaptive threshold generated by the FIR filter. For our purposes, we used the Hanning filter shown in Figure 7. The filter width, in the time domain, should be larger than the maximum frame update time of the codec under test. All points in the degraded alignment vector that are greater than the adaptive threshold generated by the filter output are declared to be peaks and only these points are included in the alignment calculation described below. The variable threshold is clipped at a heuristically set value so that this method will also be useful for non-frame-repeating codecs. This will allow use of all data points in the original and degraded alignment vectors as opposed to only the positive peaks which may occur in these vectors.

The next step in the alignment process is computation of the correlation function for all frames within the alignment uncertainty U. Many different correlation functions can be used. In practice, we have found that minimization of the variance of the difference waveform (X_n - Y_n) gives good performance. Peak output waveform values tend to have similar amplitudes to the corresponding original waveform values, and also tend to follow the same shape as the corresponding original waveform values. This is one reason why this correlation function seems to work well. The correlation function C_k is computed for each shift $k \in [0, U]$ according to the following equations:

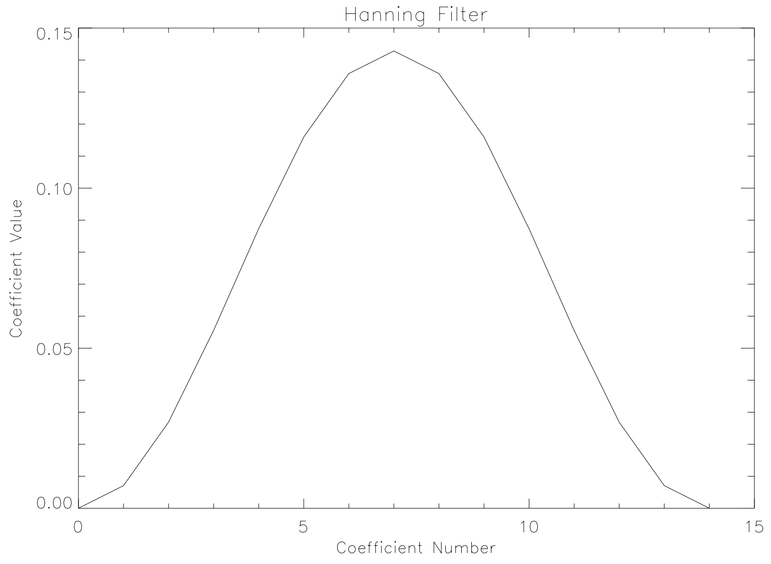


Figure 7 Hanning filter response curve.

$$m = \left(\frac{1}{N_p}\right) \sum_{j=-W/2}^{W/2} (X_{(N-k)+j} - Y_{N+j}), \forall (Y_{N+j} > f(t_{N+j})) \tag{3}$$

$$C_k = \left(\frac{1}{N_p - 1}\right) \sum_{j=-W/2}^{W/2} \{(X_{(N-k)+j} - Y_{N+j}) - m\}^2, \forall (Y_{N+j} > f(t_{N+j})) \tag{4}$$

where

N_p = total number of peaks in the degraded alignment vector, $Y_n > f(t_n)$, within W.

The shift $k_0 \in [0, U]$ for which C_k is minimum is then the best alignment. This value of k_0 then gives an estimate of the overall video delay of the transmission service channel.

Let us now consider the length of time that would be taken by the peaks algorithm in a real-time system. This algorithm can garner considerable savings because computations which are identical from one frame alignment to the next, 33 ms later in time, can be saved instead of re-computed. Specifically, the dynamic threshold need only be computed once for each degraded element. That threshold will be re-used W times, while aligning the next W frames. As a result, the dynamic threshold can be computed quite quickly.

The main time taken by the peaks algorithm is, therefore, computing the mean and standard deviation of the differences between the original and degraded elements that are greater than the threshold. This computation is executed for each of the U possible alignments. Since at least half of the degraded data points are discarded by thresholding, this step becomes on the order of $O(0.5 \cdot U \cdot W)$ steps, (W = 120 frames, U = 60 frames) which is half the order of the histogram alignment algorithm. Furthermore, computing the mean and standard deviation is less complex and therefore slightly faster than the histogram look-up required by the previous algorithm, due to the table look-up time. As a result, this algorithm takes approximately 20 ms on a 50 MHz, 486 processor, well below the real-time requirement of 33 ms.

4. Results

The performance of both of the algorithms discussed above depends upon the input scene, type of degradation, alignment window width, and alignment frame. An original scene which produces an alignment vector, $\mathrm{std}_{\mathrm{space}}(O_n - O_{n-1})$, with pseudo-periodic shape will be difficult to align as will an original scene with repetitive motion. Degraded alignment vectors, $\mathrm{std}_{\mathrm{space}}(D_n - D_{n-1})$, generated from codecs which code at 30 frames per second will align to the original more easily than those generated from codecs using frame repetition. An alignment element which falls near a motion transition point will align more easily than an element in an area with little change in the motion.

Clarification of some of our terms and ideas is pertinent before reporting the results. Alignment and delay are two different yet complementary measurements. In a real-time, causal system, the degraded image will be delayed from the associated original image by a possibly variable amount of time. Alignment of the original and degraded alignment vector elements may enable more accurate quality estimations to be obtained. Delay on the other hand is a measurement in and of itself. However, this measurement is calculated using alignment. In the block diagram of a real-time system (Figure 5), both the original and degraded alignment vector elements are time-tagged with a satellite clock. Thus, after the correct alignment between the original and degraded alignment vectors has been found by one of our alignment algorithms, the delay through the system can be calculated by comparing their time tags. Given that true delay information has been removed from our system during preprocessing, our results that will be presented in this section represent the accuracy of the delay estimate as opposed to delay itself. Thus, our plots show delay accuracy where zero is best.

The next point of clarification is that once we have decided to measure delay (or alignment), how do we verify the correctness of the measure? This is the problem of the unavailability of truth data mentioned earlier. The results were verified by a human observer viewing the original and degraded alignment vectors. If the observer was able to verify the alignment, the visual delay was subtracted from the algorithm delay results, and these points are designated with an asterisk (*) in the following plots. If the observer was unable to verify the alignment, the algorithm delay results are designated with a triangle (Δ). The original and degraded alignment vectors can be difficult for the human observer to align for several reasons: the degraded vector may have too few frame updates or bit errors, or the original vector may be erratic.

The following plots show algorithm results for degraded element 131 across our set of test clips. A clip is a given scene and degradation combination. Still and low-motion clips were filtered out for two reasons. The first is that there is not enough information in these scenes upon which to align, and the second is that delay has no significance for a still scene. Still clips were filtered out using an automated algorithm. The standard deviation of segments of the original vector of size W is calculated. If any of the standard deviations for any window W in the uncertainty region U is below a heuristically set threshold, then the clip is deemed still or low-motion and is thus filtered from the alignment program.

The final point is that the following results for the histogram method used the same set of data upon which we built the histogram table. This is because we did not have enough data to train and test independently. Thus, results for the histogram method may be overly optimistic.

Figure 8 displays results for each method over our collection of clips. Both plots in Figure 8 were generated using a 90-frame (3 sec) time window W, an uncertainty U, of 60 frames (2 sec), and aligning to degraded alignment vector element $Y_{N=131}$, which is near the center of our 9-second vectors.

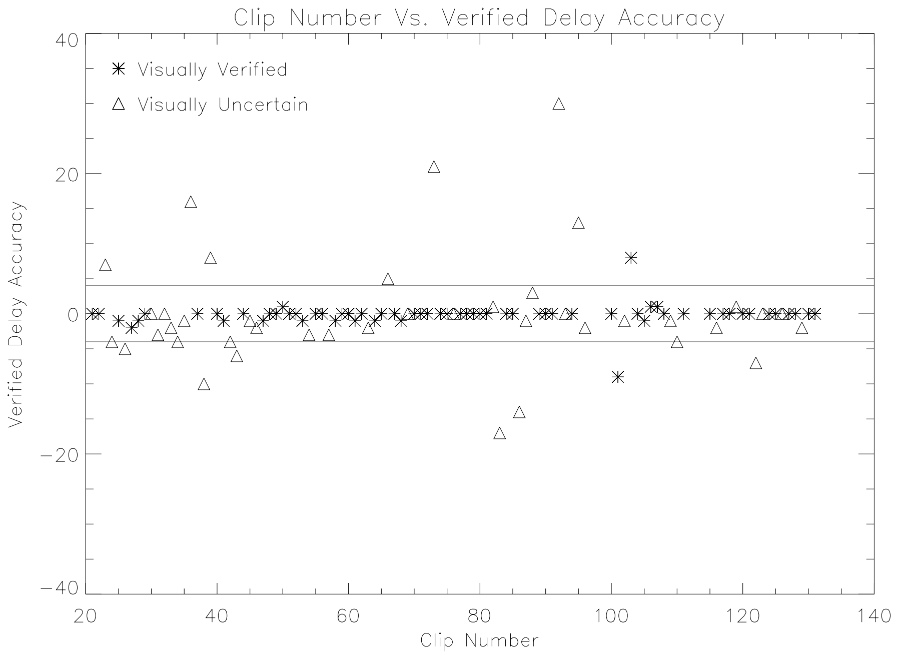
These plots illustrate that the majority of the delay accuracies fall within 0 ± 4 frames for both algorithms when the human observer was able to verify alignment. The majority of the outlying clips are clips that the human observer was unable to verify correct alignment. Thus, the automated algorithms are performing about as well as the human observer. The histogram method has more incorrectly aligned clips than the peaks method. Figure 9 presents the same information using an alignment window width W, of 150 frames (5 secs). The larger window has decreased the number of incorrectly aligned clips as expected.

Several problems affect the algorithms that do not affect a human observer. Some correlation plots have multiple peaks (maximums or minimums) which are similar in magnitude to the peak at the correct alignment point. The algorithms chose the absolute maximum (or minimum), thus ignoring similar peaks which may represent the correct alignment. Original and degraded alignment vectors that have small amounts of motion are difficult to align. Using the peaks method for low frame rate codecs, very little information may be left upon which to align. Also, data values can fall outside the range of the histograms when reading probabilities from the histogram table using the histogram method. When this occurs, the probabilities are assigned fixed values. This leads to correlation plots which have regions of constant value that make the alignment point ambiguous. These problems will be addressed in the future.

From the results we have shown here, we believe the alignment by peaks method is the most successful of the two algorithms. It has fewer clips whose delay accuracy falls outside the 0 ± 4 frame window than the histogram method. Also, for a real-time system, this method requires less computation than the histogram method.

5. Future Work

There are two areas with respect to an alignment algorithm which we feel warrant further investigation. The first is the development of a confidence measure for the delay accuracy. This would allow us to determine the likelihood that the given delay accuracy is correct. The second area of improvement we will consider is collecting statistics of the delay measurements. This will allow us to average



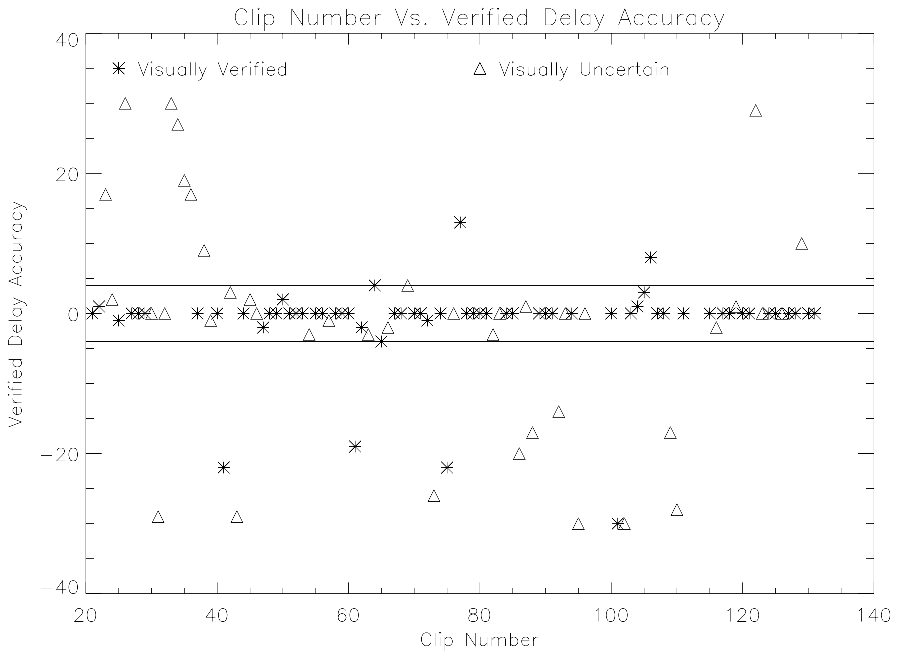
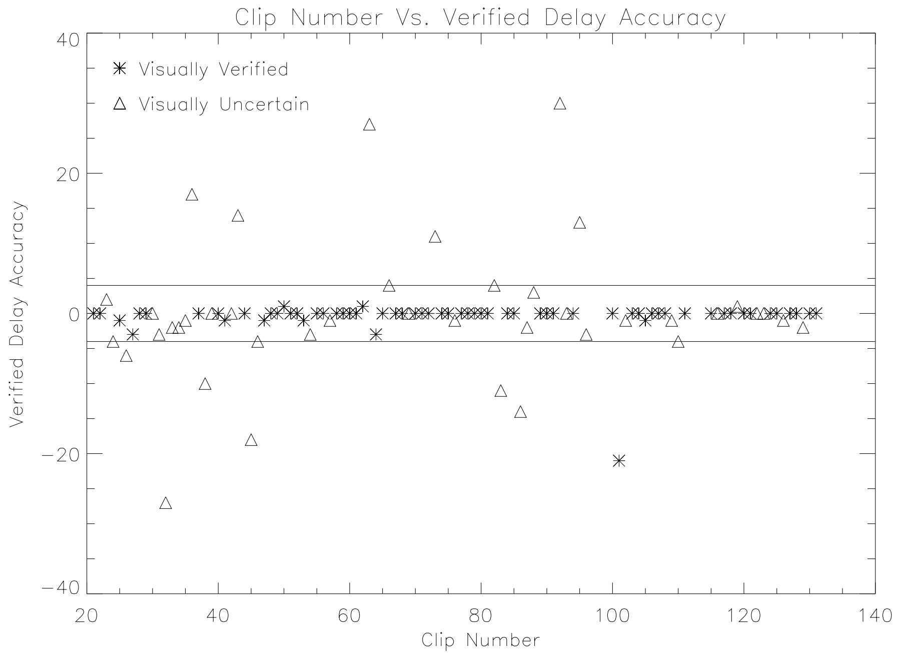


Figure 8 Alignment results across all clips. Degraded alignment vector element = 131, window width = 90 frames (3 sec), uncertainty = 60 frames (2 sec). Alignment by peaks (top), alignment by histogram (bottom).



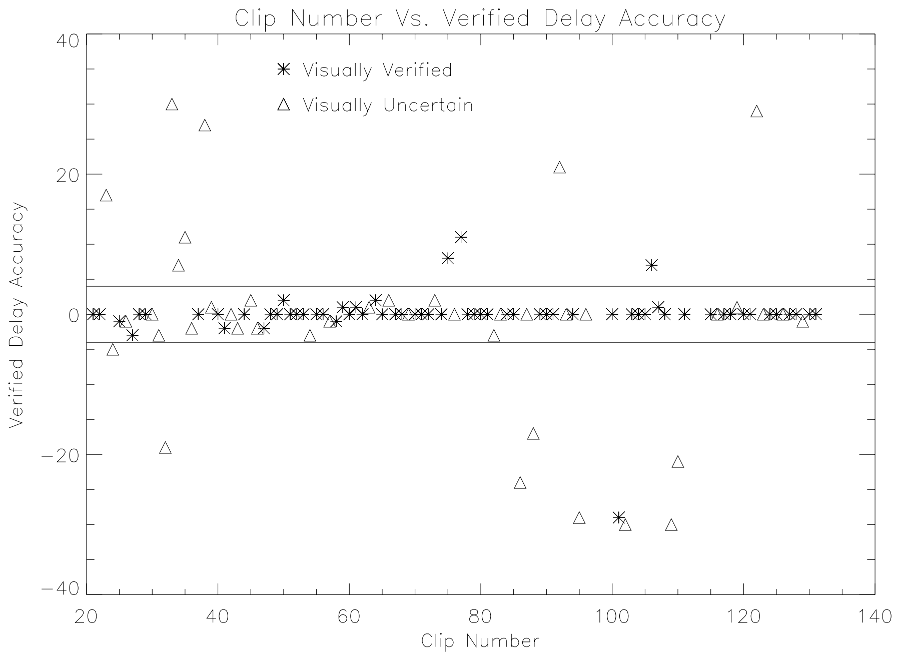


Figure 9 Alignment results across all clips. Degraded alignment vector element = 131, window width = 150 frames (5 sec), uncertainty = 60 frames (2 sec). Alignment by peaks (top), alignment by histogram (bottom).

measurements of the delay to improve accuracy. Other useful measures may also be extracted from the histogram of the one-way video delay time histories.

Similar techniques as presented here may be applicable to the audio waveform. Once audio and video delay are known, measured quantities, one can compute lip-synchronization from the difference between the video and audio delays.

We are currently in the process of developing a prototype field system for measuring objective video quality. It is intended that the system be implemented within two PC's. Figure 10 gives the block diagram of the real-time video quality measurement system. As shown in the figure, one PC will attach to the coding device, while the other will attach to the decoding device. This PC-based system will perform real-time, in-service measurements of the video quality and the one-way video delay. One of our alignment algorithms will be implemented in the box labeled "Alignment Alg.". It's input will be the mean value s_1, and mean squared value s_2, of the first order difference image. From these values, the alignment algorithm can calculate the standard deviation of the original and degraded first order difference images. This will generate the original and degraded alignment vectors X and Y used to calculate the delay. The delay will then be input to the Quality Algorithm where it will be output with the quality estimate.

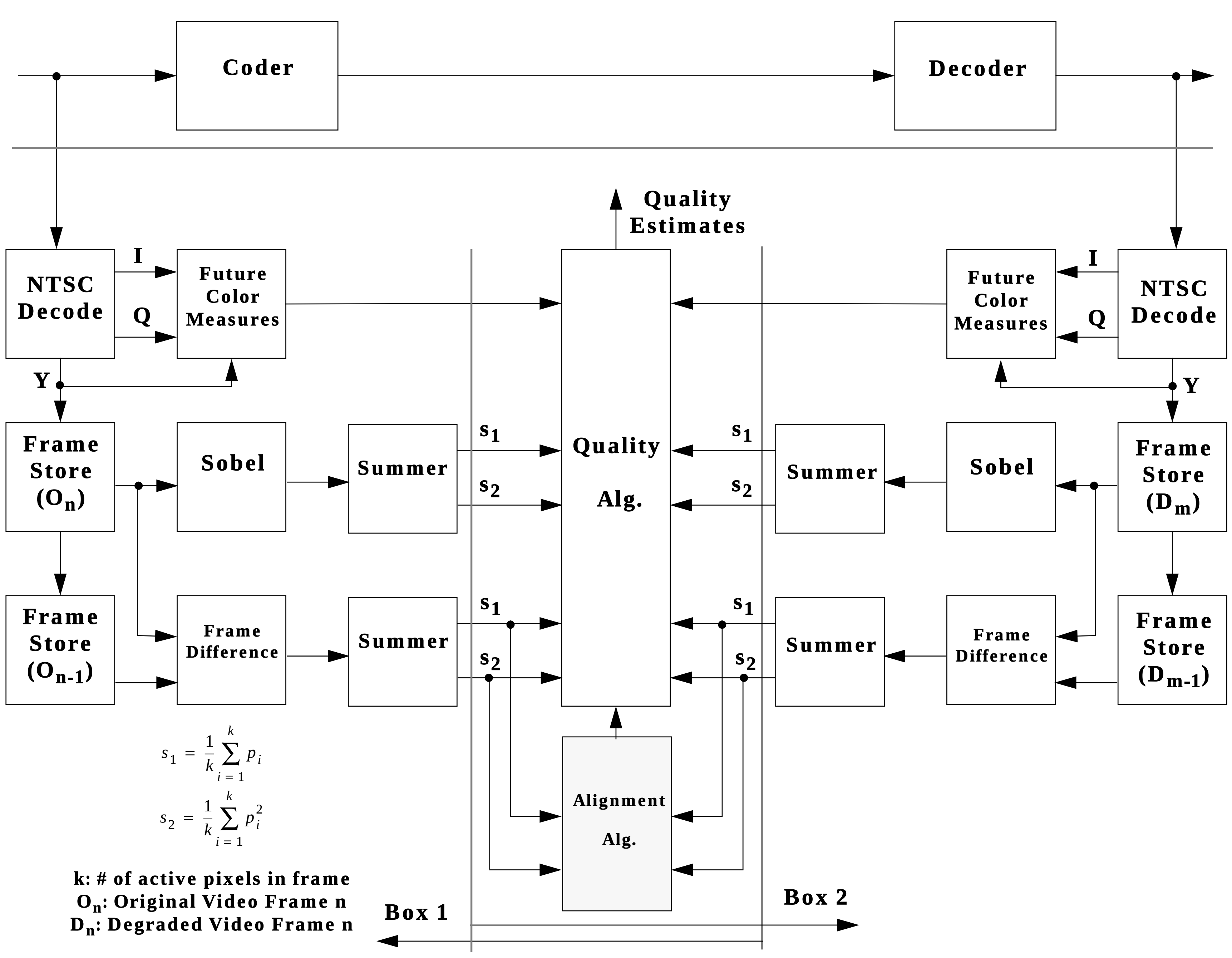


Figure 10 Real-Time Video Quality Measurement System.

Bibliography

[1] Y. Linde, A. Buzo, and R. M. Gray, “An Algorithm for Vector Quantizer Design”, *IEEE Trans. Commun.*, vol. COM-28, pp. 84-95, Jan. 1980.