International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.244

(04/2008)

SERIES J: CABLE NETWORKS AND TRANSMISSION OF TELEVISION, SOUND PROGRAMME AND OTHER MULTIMEDIA SIGNALS

Measurement of the quality of service

Full reference and reduced reference calibration methods for video transmission systems with constant misalignment of spatial and temporal domains with constant gain and offset

Recommendation ITU-T J.244

Recommendation ITU-T J.244

Full reference and reduced reference calibration methods for video transmission systems with constant misalignment of spatial and temporal domains with constant gain and offset

Summary

Recommendation ITU-T J.244 specifies full reference (FR) and reduced reference (RR) calibration methods which report alignment shifts in both space and time of processed (e.g., compressed, transmitted, and decompressed) video sequences with respect to corresponding source video sequences. The methods also compute gain and offset of processed video sequences. The methods require that spatial and temporal shifts must be constant. Furthermore, the gain and offset must be constant. The methods in this Recommendation are applicable to:

- reporting shifts that can be used to register source and processed video sequences in space and time;
- reporting gains, and offsets of video levels that can be used to calibrate processed video sequences with respect to the source.

Intended applications for these methods include, but are not limited to, assessing the operational readiness of video transmission systems that contain compression/transmission/decompression components, obtaining calibration estimates for correcting processed video sequences before measuring subjective or objective picture quality, and verifying that video transmission systems meet specified calibration requirements.

Source

Recommendation ITU-T J.244 was approved on 6 April 2008 by ITU-T Study Group 9 (2005-2008) under Recommendation ITU-T A.8 procedure.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

CONTENTS

Page

Introduction

Calibration techniques described in this Recommendation may be used for various purposes. Some of these include the verification of system setup, maintenance of delivery systems, and pre-processing for some objective video quality assessment techniques.

There has been a great interest in objective models which automatically measure video quality. For some objective quality assessment methods, the source and processed (e.g., compressed, transmitted, and decompressed) video sequences need to be registered. It is noted that some application or service scenarios may introduce spatio-temporal misalignment between processed and source video sequences. In the case where such spatio-temporal alignment problems occur, the prediction accuracy of some objective quality assessment models can be reduced. Typically, for full reference and some reduced reference objective quality assessment models, it is important that the processed and reference pictures are properly aligned. For some objective methods, alignment can be an integral part of the model. In such models there is no need for an external alignment method. For objective models that do not include inherent alignment, there may be the need for a separate picture alignment procedure that can be performed before parameter values are derived.

Intended applications for these methods include, but are not limited to, assessing the operational readiness of video transmission systems that contain compression/transmission/decompression components, obtaining calibration estimates for correcting processed video sequences before measuring subjective or objective picture quality, and verifying that video transmission systems meet specified calibration requirements.

Recommendation ITU-T J.244

Full reference and reduced reference calibration methods for video transmission systems with constant misalignment of spatial and temporal domains with constant gain and offset

1 Scope

This Recommendation specifies calibration methods which report alignment shifts in both space and time of processed (e.g., compressed, transmitted, and decompressed) video sequences with respect to corresponding source video sequences. The methods also compute gain and offset of processed video sequences. The methods require that spatial and temporal shifts must be constant. Furthermore, the gain and offset must be constant. The methods in this Recommendation are applicable to:

- reporting shifts that can be used to register source and processed video sequences in space and time;
- reporting gains, and offsets of video levels that can be used to calibrate processed video sequences with respect to the source.

These methods can be used to report calibration differences between the source and processed video sequences. These values may be used to repair processed video sequences.

Table 1 summarizes the required bandwidth and Table 2 shows the ranges of spatial and temporal shifts for which the methods in this Recommendation were tested.

Table 1 – Allowed bandwidth

	Available bandwidth for registration					
ITU-R 601(interlaced)	Unlimited	56 kbit/s	10 kbit/s	5 kbit/s	1 kbit/s	0.5 kbit/s
VGA (progressive)	Unlimited	56 kbit/s	10 kbit/s	5 kbit/s	1 kbit/s	0.5 kbit/s
CIF (progressive)	Unlimited	56 kbit/s	10 kbit/s	5 kbit/s	1 kbit/s	0.5 kbit/s
QCIF (progressive)	Unlimited	56 kbit/s	10 kbit/s	5 kbit/s	1 kbit/s	0.5 kbit/s
HD (interlaced)	Unlimited	56 kbit/s	10 kbit/s	5 kbit/s	1 kbit/s	0.5 kbit/s
HD (progressive)	Unlimited	56 kbit/s	10 kbit/s	5 kbit/s	1 kbit/s	0.5 kbit/s

Table 2 – Expected conditions

Source video length	8 seconds
Maximum spatial shift (vertical and horizontal)	±20 pixels
Maximum temporal shift	±1 second
Gain (Y-component)	±10%
Offset (Y-component)	±10 grey levels

1.1 Applications

Intended applications for these methods include, but are not limited to, assessing the operational readiness of video transmission systems that contain compression/transmission/decompression components, obtaining calibration estimates for correcting processed video sequences before measuring subjective or objective picture quality, and verifying that video transmission systems meet specified calibration requirements.

1.2 Limitations

The methods presented in this Recommendation require access to both source and processed video sequences. The spatial and temporal shifts must remain constant for the entire processed video sequences. Furthermore, the gain and offset must be fixed.

NOTE – The structure and content of this Recommendation have been organized for ease of use by those familiar with the original source material; as such, the usual style of ITU-T recommendations has not been applied.

2 References

2.1 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T J.144]	Recommendation ITU-T J.144 (2004), *Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference*.
[ITU-R BT.1683]	Recommendation ITU-R BT.1683 (2004), *Objective perceptual video quality measurement techniques for standard definition digital broadcast television in the presence of a full reference*.

2.2 Informative references

None.

3 Abbreviations

This Recommendation uses the following abbreviations:

CODEC	COder-DECoder
FR	Full Reference
GOP	Group Of Pictures
HRC	Hypothetical Reference Circuit
IRT	Institut Rundfunk Technische (Germany)
MOS	Mean Opinion Score
MPEG	Moving Picture Experts Group
NR	No (or Zero) Reference
NTSC	National Television Standard Committee (60 Hz TV)
PAL	Phase Alternating Line standard (50 Hz TV)
RR	Reduced Reference
SMPTE	Society of Motion Picture and Television Engineers
SRC	Source Reference Channel or Circuit

SSCQE Single Stimulus Continuous Quality Evaluation

VQEG Video Quality Experts Group

4 Summary of validation results

In accordance with the validation testplan for calibration methods in Appendices I and II, a total of 1190 test video sequences (525 video format) are generated from the VQEG Phase I data for various test conditions (different source video sequences and HRCs). The evaluation metrics of the testplan are provided in Appendix III. However, it was found that some of the original VQEG Phase I data were not correctly calibrated and some video sequences include still scene. Consequently, these incorrectly-calibrated and still-scene video sequences will inevitably introduce additional errors which have nothing to do with the calibration methods. When these problematic video sequences are removed, there are 973 PVSs for the 525 video format and 812 PVS for the 625 video format. Furthermore, since the test video sequences are stored in the integer format, they inherently include round-up errors which no model can predict. With those round-up errors, it would be meaningless to present the percentiles of correct results with no error in any aspect.

Tables 3-6 show the mean values of the five errors and error statistics for the 525 video format after removing the problematic video sequences. Tables 7-8 show the mean values of the five errors and error statistics for the 625 video format after removing the problematic video sequences. Appendix III presents additional error statistics.

Table 3 – Mean values of the five errors (525 format, 973 PVSs)

	Mean (Ex)	Mean (Ey)	Mean (Et)	Mean (Eg)	Mean (Eo)
NTIA 25k	0	0	–0.015	0.005	1.111
Yonsei unlimited	0	–0.00084	0.006723	0.005649	0.621204
Yonsei 56k	0	0.007563	0	0.005799	0.62936
Yonsei 10k	–0.00168	–0.00168	0.014286	0.006	0.624177
Yonsei 5k	0.00084	0.007563	0.002521	0.006184	0.590344
Yonsei 1k	0.002521	–0.00252	0.021008	0.014237	–0.56031
Yonsei 0.5k	–0.00924	0.010084	0.058824	0.01666	–0.68251

Table 4 – Error statistics of mean values (525 format, 973 PVSs)

Percentile	Ex+Ey≤0	Ex+Ey≤1	Ex+Ey+Et=0	Ex+Ey+Et≤1	Ex+Ey+Et≤2
NTIA 25k	100	100	98.739	99.748	100
Yonsei unlimited	99.91597	100	99.41176	99.83193	100
Yonsei 56k	97.22689	99.4958	96.55462	98.40336	99.91597
Yonsei 10k	99.07563	99.91597	98.23529	99.2437	99.91597
Yonsei 5k	96.38655	99.2437	94.70588	97.31092	99.4958
Yonsei 1k	95.79832	99.2437	93.61345	96.80672	99.32773
Yonsei 0.5k	95.04202	98.90756	91.84874	95.12605	97.73109

Table 5 – Supplementary error statistics of mean values (525 format, 973 PVSs)

Percentile	Et≤0	Et≤1	Et≤2	Eg≤0.01	Eg≤0.02	Eo≤2.5	Eo≤5.0
NTIA 25k	98.739	99.748	100	82.353	89.916	82.353	94.118
Yonsei unlimited	99.41176	99.91597	100	83.10924	95.71429	91.59664	98.48739
Yonsei 56k	98.23529	99.91597	100	82.26891	94.28571	90.84034	97.89916
Yonsei 10k	98.48739	99.91597	100	81.59664	93.52941	90.2521	97.89916
Yonsei 5k	96.47059	99.57983	99.7479	80.58824	92.43697	90.42017	97.47899
Yonsei 1k	95.88235	99.15966	99.66387	51.42857	75.12605	82.10084	94.78992
Yonsei 0.5k	93.94958	98.23529	98.7395	47.14286	69.15966	79.41176	93.44538

Table 6 – Mean values of the five errors (625 format, 812 PVSs)

	Mean (Ex)	Mean (Ey)	Mean (Et)	Mean (Eg)	Mean (Eo)
NTIA 25k	–0.042016807	–0.058823529	0.304201681	0.003310455	0.270601479
Yonsei unlimited	0.061345	0.018487	0.003361	0.005358	0.350661
Yonsei 56k	0.056303	0.022689	0.012605	0.005665	0.483693
Yonsei 10k	0.052941	0.026891	–0.12101	0.010251	0.901221
Yonsei 5k	0.045378	0.017647	–0.1	0.010353	0.864778
Yonsei 1k	0.086555	0.022689	–0.23109	0.022666	–1.38218
Yonsei 0.5k	0.291597	0.021008	–0.30756	0.036449	–2.527

Table 7 – Error statistics of mean values (625 format, 812 PVSs)

Percentile	Ex+Ey≤0	Ex+Ey≤1	Ex+Ey+Et=0	Ex+Ey+Et≤1	Ex+Ey+Et≤2
NTIA 25k	92.43697	98.31933	80	84.03361	90.08403
Yonsei unlimited	94.45378	99.07563	84.36975	86.89076	90.33613
Yonsei 56k	94.03361	98.82353	83.94958	86.21849	89.74790
Yonsei 10k	93.52941	98.57143	84.11765	85.96639	90.50420
Yonsei 5k	93.94958	98.48739	84.20168	86.47059	90.08403
Yonsei 1k	93.44538	97.64706	83.78151	86.05042	88.90756
Yonsei 0.5k	90.42017	95.37815	80.58824	83.94958	86.55462

Table 8 – Supplementary error statistics of mean values (625 format, 812 PVSs)

Percentile	Et≤0	Et≤1	Et≤2	Eg≤0.01	Eg≤0.02	Eo≤2.5	Eo≤5.0
NTIA 25k	80	88.82352	91.428571	88.235294	97.47899	96.63865	100
Yonsei unlimited	85.96639	90.84034	91.26050	86.63866	92.94118	93.02521	96.38655
Yonsei 56k	85.46218	90.58824	90.92437	83.44538	92.18487	92.60504	95.71429
Yonsei 10k	85.71429	91.0084	91.76471	77.05882	86.21849	85.88235	90.42017
Yonsei 5k	86.05042	90.7563	91.59664	77.22689	86.13445	86.38655	91.00840
Yonsei 1k	85.71429	90.84034	91.2605	39.41176	58.9916	64.87395	84.87395
Yonsei 0.5k	83.52941	89.83193	90.7563	32.01681	46.63866	56.55462	76.30252

5 Description of the calibration methods

The calibration methods are described in Annexes A and B.

Annex A

Institute for Telecommunication Sciences, National Telecommunications and Information Administration, U.S. Department of Commerce

Reduced Reference Video Calibration Algorithms

(This annex forms an integral part of this Recommendation)

This annex describes four Reduced Reference (RR) video calibration algorithms of low computational complexity. RR methods are useful for performing end-to-end in-service video quality measurements since these methods utilize a low bandwidth network connection between the original (source) and processed (destination) ends. The first RR video calibration algorithm computes temporal registration of the processed video stream with respect to the original video stream (i.e., video delay estimation). The second algorithm jointly calculates spatial scaling and spatial shift. The third algorithm calculates luminance gain level offset of the processed video stream with respect to the original video stream. The fourth algorithm estimates the valid video region of the original or processed video stream (i.e., the portion of the video image that contains actual picture content). All the algorithms utilize only the luminance (Y) image plane of the video signal.

Keywords: calibration; delay; gain; offset; spatial scaling; spatial shift; temporal shift; video.

A.1 Introduction

This annex describes a series of low computational complexity Reduced Reference (RR) video calibration algorithms. RR measurements are useful for in-service quality monitoring applications since they require only a small amount of reference information to make a performance measurement [1]. This RR information can thus be easily communicated between the original (i.e., source) and processed (i.e., destination) ends of a video system using commonly available network connections (e.g., Internet, Public Switched Telephone Network). In an RR measurement system, video calibration of the processed video stream (i.e., estimating gain, offset, temporal delay, spatial shift, spatial scaling, and valid region) is normally a prerequisite for estimating the quality of the processed video stream. While the algorithms described in this Recommendation can produce calibration estimates using a single video clip, these algorithms are most effective when applied in the order specified to a series of video clips associated with a single video system. Combining calibration results from multiple video clip samples provides a means to obtain more robust estimates.

The first algorithm estimates the overall temporal alignment (i.e., delay) of the processed video stream with respect to the original video stream. This temporal delay estimation algorithm utilizes three features that characterize the motion and luminance of the video scene. The second algorithm jointly estimates spatial scaling and spatial shift. This second algorithm uses randomly selected pixels and image profiles (horizontal and vertical) that are extracted from the video scene. The third algorithm reuses these pixels and profiles to estimate luminance gain and level offset. The fourth algorithm examines the edges of images and estimates the portion of the video image (original or processed) that contains actual valid picture content. All algorithms utilize the luminance (Y) image plane of the video signal. Luminance values are presumed to range from 1 to 254 (e.g., black = 16, white = 235).

A.2 Temporal registration algorithm

The temporal registration algorithm presented in this clause is an RR method that requires a data transmission bit-rate of 1 to 6 kbit/s (depending upon video frame rate and feature quantization accuracy[1]). This low bit-rate requirement, together with the computational simplicity of extracting the RR features, makes this method ideally suited for real-time in-service monitoring of end-to-end video delay. Three low bandwidth features track changes in scene motion and image intensity. Video delay is estimated by cross-correlating, or aligning, the information in the processed feature streams with the corresponding information in the original feature streams. This procedure produces one video delay estimate per video clip, which should have a minimal duration of 5 seconds[2].

A.2.1 Default valid video region

NTSC (525-line) and PAL (625-line) video sampled according to ITU-R BT.601 (henceforth abbreviated Rec. 601) may have a border of pixels and lines that do not contain valid picture. The original video from the camera may only fill a portion of the Rec. 601 frame. Some digital video compression schemes further reduce the area of the picture in order to save transmission bits. To prevent non-picture areas from degrading the performance of the temporal registration algorithm, they must be excluded.

Clause A.4 presents an automated method for determining valid region. However, since valid region is always referenced to the original video, we first need to know the spatial shift and scaling that is present in the processed video. Since spatial shift and scaling information is only available after the spatial registration algorithm is completed, and since temporal registration must be known before spatial registration, we must use some reasonable default values for valid region. Table A.1 gives the reasonable default values that we use for the border of invalid pixels around the edge of common image sizes. Pixels in this invalid border region will be discarded by the temporal registration algorithm. Images in common intermediate format (CIF), source input format (SIF), and quarter resolution versions of these (QCIF and QSIF) typically do not have an invalid border, so no pixels are discarded.

Table A.1 – Default invalid border for common video sizes

Video Type	Rows	Columns	Invalid Top	Invalid Left	Invalid Bottom	Invalid Right
NTSC (525-line)	486	720	18	22	18	22
PAL (625-line)	576	720	14	22	14	22

A.2.2 Description of Features

The temporal alignment algorithm uses three low-bandwidth features. The first temporal information (TI) feature, designated TI2, uses temporal differences between adjacent luminance images. For interlace video, fields spaced 2 fields apart in time (i.e., 1 frame apart) are differenced, and for progressive video, frames spaced one frame apart in time are differenced. This TI feature quantifies the amount of scene motion by summing the squared pixel differences between two sampled images. The second feature, designated TI10, is based on temporal differences spaced 10 interlaced fields or 5 progressive frames apart. If the frame rate in units of frames per second (FPS) decreases and the digital video system fills in the non-transmitted frames with repetitions of prior frames, processed scene motion is perceived as unnatural or discontinuous. The second feature smoothes over these times of no motion in the processed video (i.e., repeated frames or fields). The

[1] 6 kbit/s presumes interlaced video operating at 60 fields per second and single precision to code the features.

[2] The method presented here estimates the average video delay of the entire processed sequence, as opposed to methods that estimate the video delays of individual frames within the processed sequence. When individual frames within the processed sequence have different video delays, they will tend to vary about the average video delay found by this algorithm.

third feature, designated Ymean, is based on the mean of the Y images. This feature quantifies the intensity, or brightness, of the video scene by measuring the average luminance level.

Each of the three features characterizes the video differently. Thus, each works best for different amounts of frame repeating, varying video delay, and scene intensity fluctuations; and one feature may produce good temporal alignments where another one fails. The magnitude and shape of the correlation function are used as indicators for the reliability of that feature's temporal registration. Unreliable features are discarded, and the remaining results are considered jointly to estimate the best average video delay for the clip.

In the following feature definitions, the luminance image is noted as Y. For interlaced video, Y is a field and for progressive video, Y is a frame. The time when this field or frame occurs is denoted as *t*. Pixels of Y are further subscripted by row and column, *i* and *j*, respectively, so that an individual pixel is denoted as Y(*i*,*j*,*t*).

A.2.2.1 TI2 Feature: Two Field Difference Temporal Information

For interlaced video, to compute TI2 at time *t*, consider field Y(*t*) and the previous field of the same type Y(*t*–2), and compute for each pixel

$$\mathrm{TI2}(i,j,t) = \mathrm{Y}(i,j,t) - \mathrm{Y}(i,j,t-2) \qquad (1)$$

For progressive video, consider frame Y(*t*) and the previous frame Y(*t*–1), and compute for each pixel

$$\mathrm{TI2}(i,j,t) = \mathrm{Y}(i,j,t) - \mathrm{Y}(i,j,t-1) \qquad (2)$$

Then, using the results from (1) or (2), compute

$$\mathrm{TI2}(t) = rms_{\mathrm{space}}[\mathrm{TI2}(i,j,t)] \equiv \sqrt{\frac{1}{\mathrm{R}}\sum_{i}\sum_{j}(\mathrm{TI2}(i,j,t))^2} \qquad (3)$$

where rms_{space} is the root mean square function over space defined by the above equation, *i* and *j* are within the valid region defined in clause A.2.1, and R is the total number of pixels in the valid region (i.e., the number of pixels in the double summation). The TI2 calculation for interlaced fields is calculated as shown in Figure A.1.

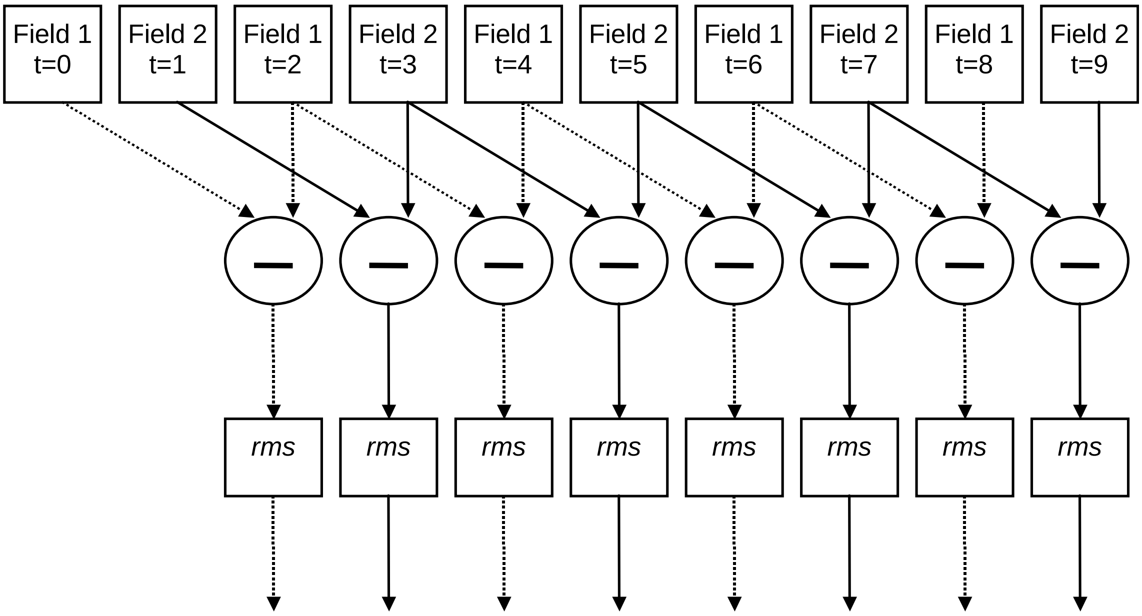


Figure A.1 – Diagram depicting the method of calculating TI2(t)

A.2.2.2 TI10 Feature: Ten field difference temporal information

The TI10 feature is based on a temporal difference spaced ten interlaced fields apart, or five progressive frames apart. This feature smoothes the temporal information using a wider filter than TI2 and eliminates the effect of frame repeats in the TI waveform for systems that have four or fewer consecutive frame repeats.

To compute TI10 at time *t* on interlaced sequences, consider field Y(*t*) and the field of the same type five frames ago Y(*t*–10), and compute for each pixel

$$TI10(i,j,t) = Y(i,j,t) - Y(i,j,t-10) \qquad (4)$$

For progressive sequences, consider frame Y(*t*) and the frame five frames ago Y(*t*–5), and compute for each pixel

$$TI10(i,j,t) = Y(i,j,t) - Y(i,j,t-5) \qquad (5)$$

Then, using the results from (4) or (5), compute

$$TI10(t) = rms_{space}\left[TI10(i,j,t)\right] \qquad (6)$$

where *i* and *j* are within the valid region defined in clause A.2.1.

A.2.2.3 Ymean feature: average luminance level

Ymean is calculated as the average luminance level of a field. To compute Ymean at time *t* on interlaced fields or frames, consider Y(*t*) and compute

$$\mathrm{Ymean}(t) = mean_{\mathrm{space}}\left[\mathrm{Y}(i,j,t)\right] \equiv \frac{1}{\mathrm{R}}\sum_{i}\sum_{j} Y(i,j,t) \qquad (7)$$

where $mean_{\mathrm{space}}$ is the mean function over space defined by the above equation, *i* and *j* are within the valid region defined in clause A.2.1, and R is the total number of pixels in this valid region (i.e., the number of pixels in the double summation).

A.2.3 Feature sequence cross-correlation[3] and validation

The following steps describe the process that is used to cross-correlate original and processed feature streams in order to estimate their best temporal registration. The algorithm includes validation steps, where the features are examined to discard potentially unreliable alignment results. The original feature stream will be referred to as a_o and the processed feature stream as a_p. For example, when using the two field difference temporal information feature, a(*t*) = TI2(*t*). Let M be the length of the feature streams, a_o and a_p. The feature stream a_o, given by {$a_o(0)$, $a_o(1)$, $a_o(2)$, …, $a_o(\mathrm{M}-1)$} will be abbreviated as $\{\mathrm{a_o}(t)\}_{t=0}^{\mathrm{M}-1}$.

We have found that temporal registration estimates based on the above features become unreliable when

$$std_{\mathrm{time}}\left[\mathrm{a_o}(t)\right] \le threshold \qquad (8)$$

or

$$std_{\mathrm{time}}\left[\mathrm{a_P}(t)\right] \le threshold \qquad (9)$$

[3] The correlation function described in this clause is based on minimization of the standard deviation of the difference between the original and processed feature streams, not the maximization of the energy in the cross-product of the two feature streams.

where *threshold* = Y_THRESHOLD = 0.25 for the Ymean feature, *threshold* = TI_THRESHOLD = 0.15 for the TI2 and TI10 features, and std_{time} represents the standard deviation over time of the M samples in the feature stream.

When (8) or (9) are satisfied, the Ymean waveform has detected insufficient temporal changes in scene brightness to be useful for temporal registration (e.g., scene with a constant brightness level). Similarly when (8) or (9) are satisfied the TI2 and TI10 waveforms have detected insufficient temporal changes in the scene motion to be useful for temporal registration (e.g., still scene). Features that fall below the thresholds in (8) or (9) are considered "invalid" and no further calculations are performed using them. Furthermore, if all three features (TI2, TI10, and Ymean) fall below the thresholds in (8) or (9), then the video clip is considered "still" and no further temporal registration calculations are performed.

Feature sequence correlation is performed on the feature streams that pass the above test. The temporal registration uncertainty, U, will be specified in *fields* for interlaced systems and *frames* for progressive systems. U represents the maximum temporal shift (plus or minus) of the processed feature stream with respect to the original feature stream. The feature sequence correlation is performed as follows:

1) Given a sequence of M processed video features $\{\mathrm{a_p}(t)\}_{t=1}^{\mathrm{M}}$, we first discard the first and last U samples to form sequence $\{\mathrm{a_p}(t)\}_{t=U}^{\mathrm{M-1-U}}$. Normalize (divide) each element in the resulting sequence by the standard deviation of that sequence to form the normalized sequence $\{\mathrm{n_p}(t)\}_{t=\mathrm{U}}^{\mathrm{M-1-U}}$.

2) For each alignment delay guess *d* (for all –U ≤ *d* ≤ U), we compute a corresponding original feature stream $\{\mathrm{a_o}(t)\}_{t=\mathrm{U+d}}^{\mathrm{M-1-U+d}}$, and normalize (divide) each element in the sequence by the standard deviation of that sequence to form the normalized sequence $\{\mathrm{n_o}(d,t)\}_{t=\mathrm{U}}^{\mathrm{M-1-U}}$. This original feature normalization is essentially the same as the processed feature normalization in step one, except computed for each delay, *d*.

3) Take the resulting normalized processed and original feature streams and compute the difference between those sequences:

$$\mathrm{Diff}(d,t) = \{\mathrm{n_o}(d,t) - \mathrm{n_p}(t)\}_{t=U}^{\mathrm{M-1-U}} \tag{10}$$

4) Compute the sample standard deviation over time of the difference sequence for each delay offset *d*, namely

$$S(d) = std_{time}(Diff\,(d,t)) \tag{11}$$

5) The minimum S(*d*) (denoted $\mathrm{S_{min}}$) and its offset $\mathrm{d_{min}}$ is the best alignment indicated by this feature. Figure A.2 gives an example plot of the correlation function S(*d*). In this plot, the best alignment occurs for the delay *d* = 0 fields (i.e., $\mathrm{d_{min}}$ = 0 fields).

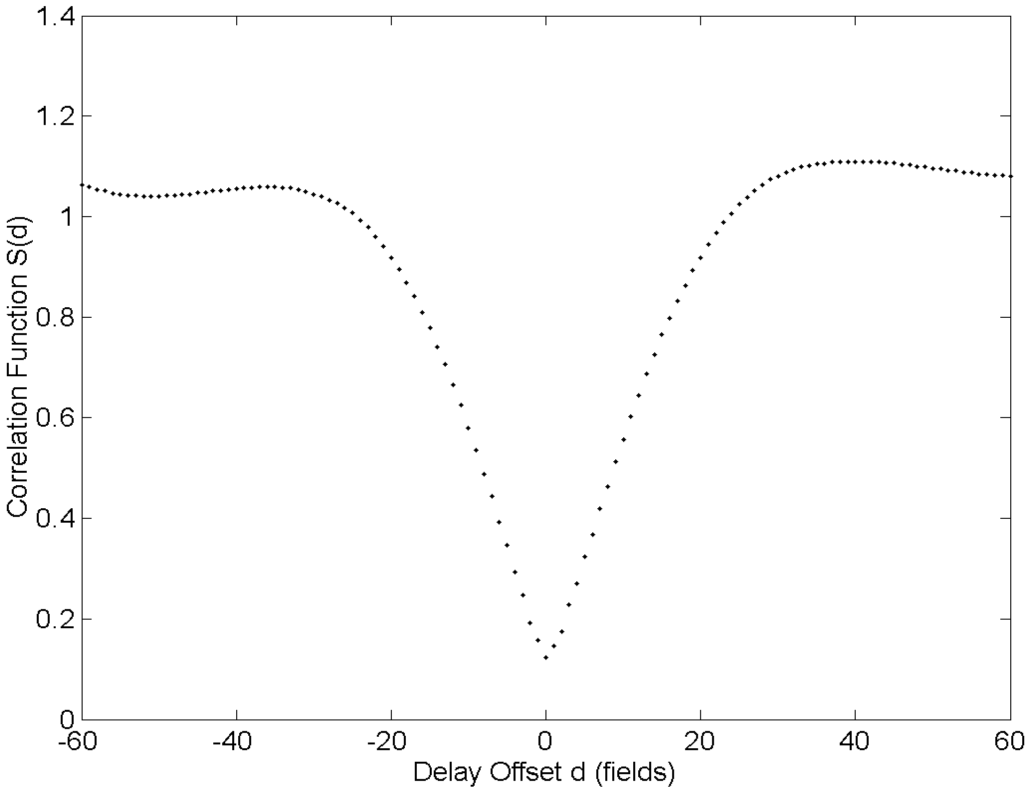


Figure A.2 – Example plot of correlation function S(*d*)

6) If the normalized original and processed feature streams were identical, then they would cancel at correct alignment (i.e., S_{min} would be 0.0). On the other hand, S_{min} can be at most 2.0, since the normalized original and processed waveforms each have a variance of 1.0. If the normalized original and processed waveforms are independent, their variances will add and S_{min} will be approximately equal to sqrt(2) ≈ 1.414. A value for S_{min} greater than sqrt(2) indicates that the two waveforms are negatively correlated. We have found that if $S_{min} \leq$ CORRELATION_VALID = 0.25, then the correlation between the processed feature stream and the original feature stream is probably reliable and these features are therefore considered valid. However, if $S_{min} \geq$ CORRELATION_INVALID = 1.40, then the correlation between the processed feature stream and the original feature stream is unreliable for the above reasons and these features are therefore considered invalid. For correlations in-between (e.g., CORRELATION_VALID ≤ S_{min} ≤ CORRELATION_INVALID), S_{min} yields ambiguous results with respect to accuracy, so other criteria must be used (see step 7).

7) If (CORRELATION_VALID ≤ S_{min} ≤ CORRELATION_INVALID), find the earliest (minimum) delay, $d_{1,}$ where S(d_1) < S_{min} + DELTA_THRESHOLD, where DELTA_THRESHOLD = 0.04. Also find the latest delay, $d_{2,}$ where S(d_2) < S_{min} + DELTA_THRESHOLD. Compute the distance between those two delays, $w = d_2 - d_1 + 1$. Notice that no restrictions are placed on values of S for delays between d_1 and d_2. This width *w* discriminates between correlations with a well-defined minimum, and correlations with multiple nearly-identical minimum values (e.g., a sharp correlation function as given in Figure A.2 versus a broad correlation function). TI features with *w* ≤ TI_WIDTH are reliable, and Ymean features with *w* ≤ Y_WIDTH are reliable, where TI_WIDTH = 3 and Y_WIDTH = 4. Features that meet these criteria are considered valid whereas features that do not meet these criteria are considered invalid.

Table A.2 provides a summary of the recommended threshold values that are used by the temporal registration algorithm. These recommended threshold values were obtained by minimizing alignment errors between the current algorithm and the frame-based temporal alignment method in clause 3.4.2 of [3].

Table A.2 – Recommended values for thresholds used by temporal registration algorithm

Threshold	Recommended Value
TI_THRESHOLD	0.15
Y_THRESHOLD	0.25
CORRELATION_VALID	0.25
CORRELATION_INVALID	1.40
DELTA_THRESHOLD	0.04
TI_WIDTH	3
Y_WIDTH	4

A.2.4 Estimation of Temporal Offset from Features

The following describes how to apply the three features of clause A.2.1 and the correlation algorithm of clause A.2.3 to achieve the final estimate of the temporal alignment offset.

1) Compute the original and processed TI2 feature streams, TI10 feature streams, and Ymean feature streams in clause A.2.1. When operating in-service, transmit the original features to the processed video location. At most, single precision will be required (i.e., 4-byte floating point), and further bandwidth savings can be obtained through quantization (e.g., 16 bits per value).

2) For each original and processed feature stream pair, compute the correlation function S(*d*) according to clause A.2.3 and record whether that feature is valid or invalid, and if invalid whether that feature is still (e.g., motionless).

3) If one or more features are valid, average those valid correlation functions together. For progressive video sequences, find the delay offset S_{min} that minimizes the averaged correlation function. For interlaced video sequences, this algorithm may be run with either field or frame accuracy. For frame accurate delay, restrict the search to either field one delays or field two delays; for field accurate delay, include both field one and field two delays in the search. Find the delay offset S_{min} that minimizes the averaged correlation function.

NOTE – This algorithm cannot be used to determine with 100% reliability whether field one of the processed video sequence best aligns to field one or field two of the original sequence (i.e., indicative of interlaced reframing by the video system under test – see clause 3.1.2 of [3]). We have found that such a determination is approximately 90% accurate. Thus, to detect reframing we recommend the use of some other, external algorithm such as the one that will be described in clause A.3. Field accurate delays from this temporal registration algorithm provide rough estimates, suitable when subsequent steps will improve those estimates.

4) If all of the features are invalid, a delay cannot be computed for this video clip. Furthermore, if all features have been marked still, then the clip contains still video and this extra information may be reported to the user. Temporal alignment or registration is not required to estimate quality for still scenes.

A.2.5 Observations and Conclusions

The in-service video delay measurement algorithm presented here uses a set of very low bandwidth features extracted from original and processed video sequences. This algorithm is suitable for measuring video delay in a fully automated, in-service, real-time monitoring system or for aligning video in an out-of-service environment, prior to performing video quality measurements. The video delay measurement algorithm utilizes two types of features, those that track changes in scene motion (TI2, TI10), and those that track changes in scene intensity (Ymean). The video delay measurement algorithm provides good estimates of video delay for a wide range of video scenes and systems by correlating, or time aligning, these original and processed sampled feature values.

When field or frame repeats are present in the processed video sequence, the definition of video delay is somewhat ambiguous. Does a viewer perceive video scene delay as the delay of the first processed field in the field repeat sequence, the last processed field in the field repeat sequence, or something in between? Because the three features operate at different levels of temporal granularity, the video delays measured by these features may differ slightly for video systems that have field or frame repeats. By combining the results from all three features, a more robust delay estimate is obtained.

The length of the time window that is used to estimate delay limits the response of the measurement system to changes in video delay. Shorter time windows can cause measurement errors when the video contains small but perceptible amounts of repetitive motion limited to a small portion of the image (e.g., a four second head and shoulders scene wherein the only change in scene content is due to lip motion). Shorter time windows can be reliably used when the video contains variations in image intensity and/or scene motion. As the length of the time window increases, the measurement system will respond more slowly to variations in video delay.

This temporal registration algorithm is robust when other calibration errors are present. For instance, the temporal registration estimates are not significantly impacted by video system luminance gain and level offset errors. This was verified by inserting random luminance gains from 0.8 to 1.2 and random luminance level offsets from –20 to +20 into 2772 processed video sequences. The resulting temporal registration values were unchanged.

This temporal registration algorithm is only minimally impacted by spatial shifts and spatial scaling. This was tested by inserting random spatial scaling from 5% shrinkage to 5% expansion in both the horizontal and vertical directions, while simultaneously inserting random spatial shifts from –20 to +20 both horizontally and vertically. Figure A.3 compares the temporal registration results from 2772 processed video sequences containing these spatial calibration errors with the temporal registration results from fully calibrated processed video. Notice that most of the video clip's alignments agree within plus or minus one frame, even in the presence of extreme calibration errors. Since many of the video clips contained frame repeats, some of the differences between running this temporal registration algorithm on calibrated and un-calibrated video clips may be indicative of truly ambiguous temporal registration. Spatial calibration errors increased the number of video clips with all features invalid from 159 to 165.

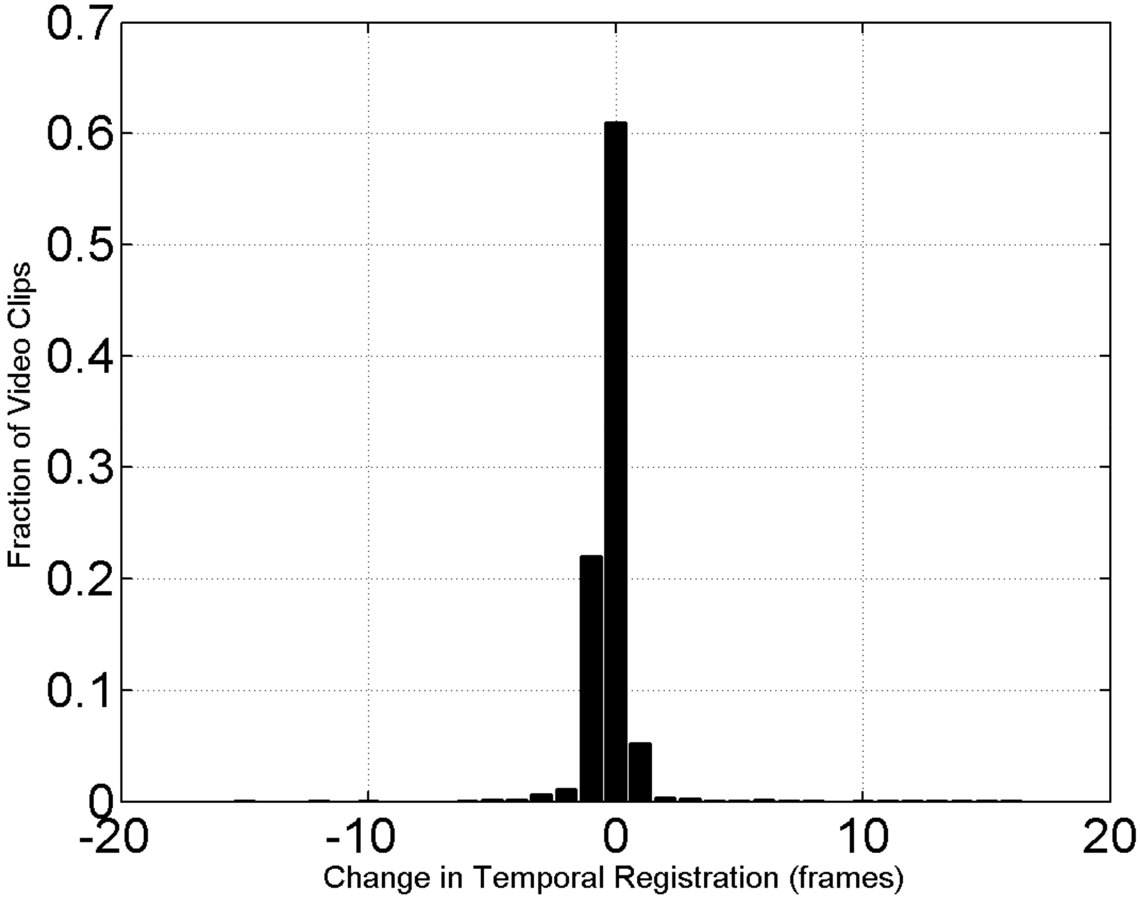


Figure A.3 – Impact of spatial calibration errors on RR temporal registration

A.3 Spatial registration algorithm

The spatial registration algorithm requires as input the temporal offset between the original and processed video streams. This temporal offset is obtained from the temporal registration algorithm in clause A.2. The spatial registration algorithm simultaneously estimates horizontal and vertical spatial scaling and spatial shift. This searching utilizes RR features that are extracted from one frame every second. All used RR features of the video clip are considered simultaneously, to enhance the estimation accuracy. The spatial registration algorithm requires an RR data transmission bit rate of 4 to 30 kbit/s (depending upon image size and feature quantization accuracy).

The algorithm presented here is a downstream RR algorithm. Low bandwidth RR information from the original video sequence is transmitted to the processed video sequence location. More information is required from the processed video sequence than the original video sequence to perform the spatial searches. This algorithm can also be implemented as an upstream monitoring system with no loss in accuracy (i.e., the RR information is transmitted from the processed end to the original end, and extra information is used from the original video sequence rather than from the processed video sequence).

A.3.1 Preconditions

For the spatial registration algorithm to operate reliably, the original and processed video sequences must be temporally aligned. Since the temporal registration algorithm described in clause A.2 is robust against spatial shift and scaling errors that are present in the processed sequence, it may be used to temporally align the processed video before proceeding with spatial registration. The output of temporal registration is a single estimate of delay between the original and processed video sequences. For processed video sequences that exhibit constant or variable frame rate encoding, the temporal registration algorithm will produce an estimate of the average video delay for the clip.

The spatial registration algorithm does not perform *any* temporal search of its own. Temporal registration using the estimated average delay is presumed to be sufficiently accurate. This approach was found to yield accurate results for processed video sequences exhibiting constant or variable frame rate encoding. The training data used to develop the spatial registration algorithm included transmission errors, and these errors produced pausing with loss, pausing without loss, and corrupted images. The training data also included a wide range of frame rates, including variable frame rate encoding that occasionally was slower than 1 fps.

Because this algorithm does not do a temporal search, it is able to treat progressive and interlaced video sequences identically. Complications resulting from interlaced field reframing only arise when field streams are slid past each other. The elimination of temporal registration search from the spatial registration algorithm eliminates the need to treat interlaced video differently from progressive video.

There is one minor difference in how interlaced video frames are read, if and only if field accurate temporal registration indicates reframing. In this case, when reading a processed video frame, the earlier field in time should be replaced with the earlier field from the next frame. For example, for lower-field-first processed video with a delay of one field (with respect to the original), replace the lower field of the current frame with the lower field from the next frame. Notice that this differs substantially from reframing (i.e.., moving the later field of the current frame into the earlier field, and moving the earlier field of the next frame into the later field of the current frame), which requires both a spatial and a temporal shift. The replacement of the earlier field as described here has no impact on the spatial shift or scaling results, whereas reframing would affect the measured vertical spatial shift by 1 frame line.

As with temporal registration, a border of pixels and lines that do not contain valid picture should be eliminated (see clause A.2.1).

A.3.2 Algorithm overview

A search over all of these available pixels in space and time would be prohibitively time consuming. Additionally, preliminary results indicate that these exhaustive searches can produce inaccurate results. Distortions present in the processed video sequence can result in erroneous spatial shift/scale values being selected. Instead, this spatial registration algorithm uses three types of RR information extracted from one frame every second: 1) a small number of pixels chosen at random, 2) the horizontal profile of pixels (average value of each column), and 3) the vertical profile of pixels (average value of each row). Due to averaging, the profiles are less susceptible to noise and thus provide a robust estimation of spatial shift and scaling, but cannot give a precise estimate. The randomly selected individual pixels yield more precise estimates, but are less robust. Taken together, the profiles and pixels yield robust and precise estimates of spatial shift and scaling.

Before continuing, we would like to explain the use of randomized pixel selection. Intuitively, it seems appropriate to use a heuristic to intelligently select pixels that are well suited to the task. However, it is quite difficult to develop a heuristic that provides an accurate response to the infinite variety of scene content and system impairments. During algorithm development, many heuristics were tried but none proved as robust as random choice. For example, pixels near strong edges are appropriate for some content, but misleading for other content. When selecting original pixels, it is difficult to guess the portions of the image that will be heavily impaired (and thus less useful) and the portions that will be accurately rendered – this depends too much upon the inner workings of the unspecified coding and transmission scheme. Selecting pixels randomly provides a Monte Carlo method for finding a near optimal solution.

A.3.3 Scaling and shift algorithm

The following constants are used in the spatial registration algorithm. HSHIFT is the maximum amount of horizontal shift to be searched, and VSHIFT is the maximum amount of vertical shift to be searched. These constants are specified in pixels for horizontal shift, and frame-lines for vertical shift. Both must be even numbers. HSCALE is the maximum horizontal scaling to be searched, and VSCALE is the maximum vertical scaling to be searched. These constants divided by 1000 yield the maximum fraction of image expansion or shrinkage. Thus, 60 indicates the expansion or shrinkage is limited to 6% (i.e., the scale factor is in the range 94% to 106%). Larger or smaller values can be used for both of these constants where appropriate. Table A.3 gives reasonable default values of these constants for common image sizes.

Table A.3 – Default HSHIFT, VSHIFT, HSCALE, and VSCALE values

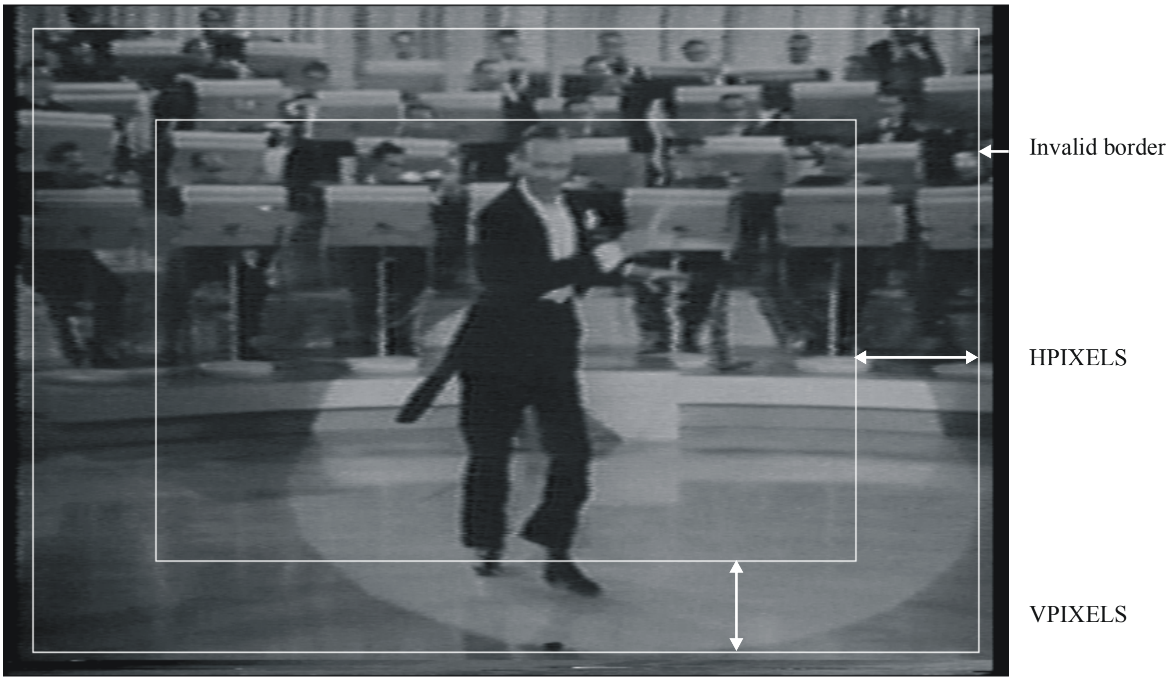
Resolution	QCIF, QSIF	CIF, SIF	VGA, NTSC, PAL
HSHIFT	4	8	20
VSHIFT	4	8	20
HSCALE	60	60	100
VSCALE	60	60	100

The original RR video information that is transmitted must not come from pixels near the border of the valid region, as those pixels in the processed video are required for scaling and shifting within searches. To eliminate these pixels, an additional border is required of width given by:

$$\text{HPIXELS} = \textit{evenup}\ (\text{HSHIFT} + (\text{HSCALE}/1000) * \textit{cols}) \tag{12}$$

$$\text{VPIXELS} = \textit{evenup}\ (\text{VSHIFT} + (\text{VSCALE}/1000) * \textit{rows}) \tag{13}$$

where *rows* and *cols* are the number of rows and columns in the original image after eliminating the invalid border given in Table A.1. Function *evenup* rounds a value up to the next even integer. For NTSC/525-line using the above suggested values, *rows* = 450, *cols* = 676, and thus HPIXELS = 88 and VPIXELS = 66 as shown in Figure A.4.



J.244(08)_FA.4

Figure A.4 – HPIXELS and VPIXELS visually demonstrated

For the entire original video clip (e.g., 8 to 15 seconds long), the images to be used by the scaling and shift algorithms are first selected. It is recommended that the selected images be uniformly sampled in time and spaced one second apart. Thus, for an *N* second long sequence, *N* images are chosen. Let Y_n be the *n*th luminance image in an original video clip that contains these *N* images (i.e., $n = 1, 2, \ldots, N$). Let $Y_n(i,j)$ be the coordinates of a pixel, where *i* is the vertical row index and *j* is the horizontal column index, and the upper-left coordinate of the image is $i = 1, j = 1$. Eliminate the invalid border shown in Table A.1 and also a border of HPIXELS on left and right, and VPIXELS on top and bottom. Let O_n be the resulting image. For our NTSC/525-line example, O_n now contains 318 rows and 500 columns.

Compute the vertical profile of each original image (i.e., average each row) and join the profiles together into a single profile array, $VP_O(i,n)$, given by

$$VP_O(i,n) = \frac{1}{C_o}\sum_{j=1}^{C_o} O_n(i,j) \tag{14}$$

where C_o is the total number of columns in O_n.

Compute the horizontal profile of each original image (i.e., average each column) and join the profiles together into a single profile array, $HP_O(j,n)$, given by

$$HP_O(j,n) = \frac{1}{R_o}\sum_{i=1}^{R_o} O_n(i,j) \tag{15}$$

where R_o is the total number of rows in O_n.

A random number sequence will now be used to select random pixels from the *N* original video images O_n. Let *S* be an integer in the range [0, 255], chosen at random. *S* will be used as a seed for the pseudo-random number generator and can be transmitted in one byte. Since each computer uses the same seed *S* and the same pseudo-random number generator, each computer will produce the same random number sequence. As given by equation (16) below, we will use 80% as many random individual pixels as profile values. This 80% number is based on empirical evaluations of the algorithm performance and attaches slightly more weight to the horizontal and vertical profile samples versus the random pixel samples. Let M be the number of individual pixels to be selected from the video sequence.

$$M = round\ (0.8 * N * (R_o + C_o)) \tag{16}$$

$$orig\_i = round\ (rand(M) * R_o + 0.5) \tag{17}$$

$$orig\_j = round\ (rand(M) * C_o + 0.5) \tag{18}$$

$$orig\_n = round\ (rand(M) * N + 0.5) \tag{19}$$

where $rand(M)$ is a vector of M numbers from a random number generator with a uniform distribution over the range (0, 1) and *round* is a function that rounds to the nearest integer. Arrays *orig_i, orig_j, and orig_n* contain the row, column, and time coordinates for each of the M pixels, respectively. Each pixel location is chosen independently. Thus, more pixels may be selected from some images and less pixels from other images. For our NTSC/525-line example, *round* $(0.8 * N * 818)$ pixels will be selected.

The next task is to find the best combination of horizontal scaling, vertical scaling, horizontal shift and vertical shift. An exhaustive search over these four dimensions would be prohibitively time consuming. Therefore, a randomized, iterative search strategy is used instead.

We begin by defining our initial guesses. Let *hm* be the current horizontal scaling factor (multiplicative), *ha* the current horizontal shift factor (additive), *vm* the current vertical scaling factor (multiplicative), and *va* the current vertical shift factor (additive). At all times,

$$\begin{gathered} -\text{HSHIFT} < ha < \text{HSHIFT} \\ -\text{VSHIFT} < va < \text{VSHIFT} \\ -\text{HSCALE} < hm < \text{HSCALE} \\ -\text{VSCALE} < vm < \text{VSCALE} \end{gathered} \tag{20}$$

For each of the N luminance images in the processed video sequence, eliminate the invalid border shown in Table A.1. Let P_n be the resulting processed image, R_p the number of rows in that image and C_p the number of columns. Notice that $R_\text{p} = R_o + 2 *$ VPIXELS; and $C_p = C_o + 2 *$ HPIXELS. Recall that P_n and O_n have already been temporally aligned by some external temporal registration algorithm such as that presented in clause A.2. For our NTSC/525-line example, P_n contains 450 rows (R_p) and 676 columns (C_p).

One could scale the processed video images by factors *hm* and *vm* before extracting the horizontal and vertical profiles. However, that approach is not very computationally efficient as it requires multiple two-dimensional resamplings of the processed images. Instead, the horizontal and vertical profiles are extracted from the processed video images and these extracted profiles are rescaled by factors *hm* and *vm*.

Compute the vertical profile of each processed image (i.e., average each row) and join the profiles together into a single profile array, $VP_p(i,n)$, given by

$$VP_p(i,n) = \frac{1}{C_p}\sum_{j=1}^{C_p} P_n(i,j) \tag{21}$$

Compute the horizontal profile of each processed image (i.e., average each column) and join the profiles together into a single profile array, $HP_p(j,n)$, given by

$$HP_p(j,n) = \frac{1}{R_p}\sum_{i=1}^{R_p} P_n(i,j) \tag{22}$$

To further speed computations, rescaling of the processed horizontal and vertical profiles is performed using nearest neighbour interpolation. We will first scale and then shift the original coordinates according to equations (23) and (24) below, and these scaled coordinates will then be used to index the processed profiles calculated by equations (21) and (22). The new vertical and horizontal profile indices (*prof_i* and *prof_j*, respectively) are given by

$$prof\_i = round\left(\frac{i}{vms+1} + va + \text{VPIXELS} + \frac{vms}{vms+1}\frac{R_o}{2}\right),\ i = 1, 2, \ldots, R_o \tag{23}$$

$$prof\_j = round\left(\frac{j}{hms+1} + ha + \text{HPIXELS} + \frac{hms}{hms+1}\frac{C_o}{2}\right),\ j = 1, 2, \ldots, C_o \tag{24}$$

where

$$vms = \frac{vm}{1000},\ hms = \frac{hm}{1000} \tag{25}$$

Similarly, the scaled and shifted image coordinates for the randomly sampled pixels are:

$$proc\_i = round\left(\frac{orig\_i}{vms+1} + va + \text{VPIXELS} + \frac{vms}{vms+1}\frac{R_o}{2}\right) \tag{26}$$

$$proc\_j = round\left(\frac{orig\_j}{hms+1} + ha + \text{HPIXELS} + \frac{hms}{hms+1}\frac{C_o}{2}\right) \tag{27}$$

$$proc\_n = orig\_n \tag{28}$$

Since the processed and original images are assumed to have been temporally aligned, the time index *orig_n* does not need to be scaled or shifted and thus remains unchanged.

The next task is to jointly consider the randomly chosen pixels, the vertical profiles, and the horizontal profiles. Let $\underline{O}$ be a column vector containing a concatenation of the original pixels from O_n at locations (*orig_i*, *orig_j*, *orig_n*), the original vertical profiles VP_o for $n = 1, 2, \ldots, N$, and the original horizontal profiles HP_o for $n = 1, 2, \ldots, N$. Similarly, let $\underline{P}$ be a column vector containing a concatenation of the processed pixels from P_n at locations (*proc_i*, *proc_j*, *proc_n*), the processed vertical profiles VP_p at locations *prof_i* for $n = 1, 2, \ldots, N$, and the processed horizontal profiles HP_p at locations *prof_j* for $n = 1, 2, \ldots, N$.

Let us now define our search criteria, *V*, which depends upon *hm*, *ha*, *vm* and *va*. *V* is the standard deviation of the difference between $\underline{O}$ and $\underline{P}$, namely

$$V = stdev(\underline{O} - \underline{P}) \tag{29}$$

Smaller values of *V* indicate a closer match between original and processed.

The search strategy contains two stages. The first stage searches randomly and with uniform density across the entire search space. The second stage refines the results of the first stage. It uses a 4-dimensional Gaussian distribution to focus the search in the vicinity of the current best point that minimizes *V* in equation (29). Each time a new best point is found, the second stage search is recentred about that point.

Let us define six variables: *W, min_W, min_hm, min_ha, min_vm,* and *min_va*. *W*(*hm, ha, vm, va*) will hold *V* for each horizontal scale *hm*, horizontal shift *ha*, vertical scale *vm*, and vertical shift *va*.

$W(hm, ha, vm, va)$ is initialized to NaN (Not-A-Number). min_W will hold the minimum V, whose value will be associated with horizontal scale min_hm, horizontal shift min_ha, vertical scale min_hm, and vertical shift min_ha. Initialize min_W to infinity. Note that hm will range from –HSCALE to HSCALE, while ha will range from –HSHIFT to HSHIFT. Likewise, vm will range from –VSCALE to VSCALE, while va will range from –VSHIFT to VSHIFT. Finally, let us choose TRIES, the number of evaluations to be performed before the algorithm declares that a solution has been found. A value of TRIES = 15000 seems to work well and is the recommended default value.

For a number of iterations equal to TRIES/10, choose values for hm, ha, vm, and va randomly over the range to be searched, using a uniform distribution of random values.

$$hm = round\,(-\mathrm{HSCALE} + ((\mathrm{HSCALE} * 2) * rand)) \tag{30}$$

$$ha = round\,(-\mathrm{HSHIFT} + ((\mathrm{HSHIFT} * 2) * rand)) \tag{31}$$

$$vm = round\,(-\mathrm{VSCALE} + ((\mathrm{VSCALE} * 2 * rand)) \tag{32}$$

$$va = round\,(-\mathrm{VSHIFT} + ((\mathrm{VSHIFT} * 2) * rand)) \tag{33}$$

where $rand$ is a random number generator that yields numbers from the uniform distribution over the range (0, 1).

For each randomly chosen coordinate (hm, ha, vm, va), compute V as shown in equation (29) which will give the value for $W(hm, ha, vm, va)$. Update the values of $W, min\_W, min\_hm, min\_ha, min\_vm, min\_va$ in equations (34) and (35).

$$W(hm, ha, vm, va) = V \tag{34}$$

If $V < min\_W$, then

$$min\_W = V,\ min\_hm = hm,\ min\_ha = ha,\ min\_vm = vm, \text{ and } min\_va = va \tag{35}$$

If a coordinate (hm, ha, vm, va) is chosen twice, then the calculation of V is skipped. Duplicate coordinates are detected by testing whether $W(hm, ha, vm, va)$ contains NaN. Duplicate coordinates are counted in the number of evaluations to be tried.

After the TRIES/10 = 1500 iterations, the coordinate $(min\_hm, min\_ha, min\_vm, min\_va)$ will be a fairly close estimate of the actual coordinate of the minimum of V. Perform an additional TRIES * 9/10 iterations as shown above but with a modified distribution of random values. The new random distribution increases the likelihood of the chosen coordinate being closer to the current best point in the search space.

$$hm = min\_hm + round\,(2 * rand\_norm) \tag{36}$$

$$ha = min\_ha + round\,(2 * rand\_norm) \tag{37}$$

$$vm = min\_vm + round\,(2 * rand\_norm) \tag{38}$$

$$va = min\_va + round\,(2 * rand\_norm) \tag{39}$$

In equations (36)-(39), $rand\_norm$ is a random number generator that yields a normal distribution with zero mean and unity variance. If a random coordinate (hm, ha, vm, va) is outside the range to be searched, then another random coordinate is chosen instead. The long tails of the normal distribution help prevent the algorithm from locking in on a local minimum rather than the global minimum. The quick handling of duplicate coordinates allows TRIES to be set to a large number without negatively impacting run speed. Note that equations (36)-(39) continually recenter the search about the current best point in the search space.

The final values for *min_hm, min_ha, min_vm,* and *min_va* indicate the horizontal and vertical scaling and shift factors. The *min_hm* and *min_vm* spatial scaling factors are used by a resampling routine to resample the processed video images[4]. After this spatial scaling, the resultant images are then shift corrected by *min_ha* and *min_va*. The user should verify that the processed video after calibration is properly registered to the original video.

A.3.4 Median Filtering Values from Multiple Video Sequences

Spatial shift and scaling results for individual processed video clips contain occasional outliers. Still or nearly still scenes are particularly problematic, since estimates are essentially based upon one image rather than a sequence of images. If multiple processed video sequences that have passed through the same video system (i.e., all video sequences can be considered to have the same calibration numbers, except for temporal registration) are available, then spatial shift and spatial scaling results should be filtered across scenes to achieve increased accuracy. We have found that median filtering across scenes produces robust estimates for spatial scaling and spatial shift. If possible, we recommend median filtering across results from seven or more different video scenes.

Spatial shift was verified by comparing the results for 2565 processed video clips for this algorithm and the frame-based spatial shift algorithm in clause 3.1.5 of [3]. The number of clips for the spatial registration validation (2565) was different from the number of clips for the temporal registration validation (2772) because spatial registration errors were introduced into some clips from tape editing, and these clips were discarded. Each video system represented by those 2565 video clips has at least two different original video clips, so that a limited amount of median filtering using different scenes could be performed for this study. Figure A.5 shows the distance between the spatial shifts produced by this algorithm and those produced by the frame-based spatial registration algorithm in clause 3.1.5 of [3]. The plot in Figure A.5 is for individual clips and does not include any median filtering over scenes to increase accuracy.

Spatial scaling was verified by comparing the results for the above video clips to the spatial scaling algorithm in [4]. Figure A.6 shows the distance between the spatial scaling of this algorithm and those produced by the algorithm in [4]. These spatial scaling results were visually verified when scaling was detected as being present. The plot in Figure A.6 is for individual clips and does not include any median filtering over scenes to increase accuracy.

Figure A.7 shows the increased agreement between the two sets of spatial shift results given in Figure A.5 when median filtering over scenes is performed for both algorithms. Similarly, Figure A.8 shows the increased agreement between the two sets of spatial scaling results given in Figure A.6. For Figures A.7 and A.8, there were a total of 247 different video systems. Notice that the median filtered results are tightly clustered around zero.

[4] This report makes no recommendations on how to perform high accuracy resampling of the processed video images using the final spatial scaling factors.

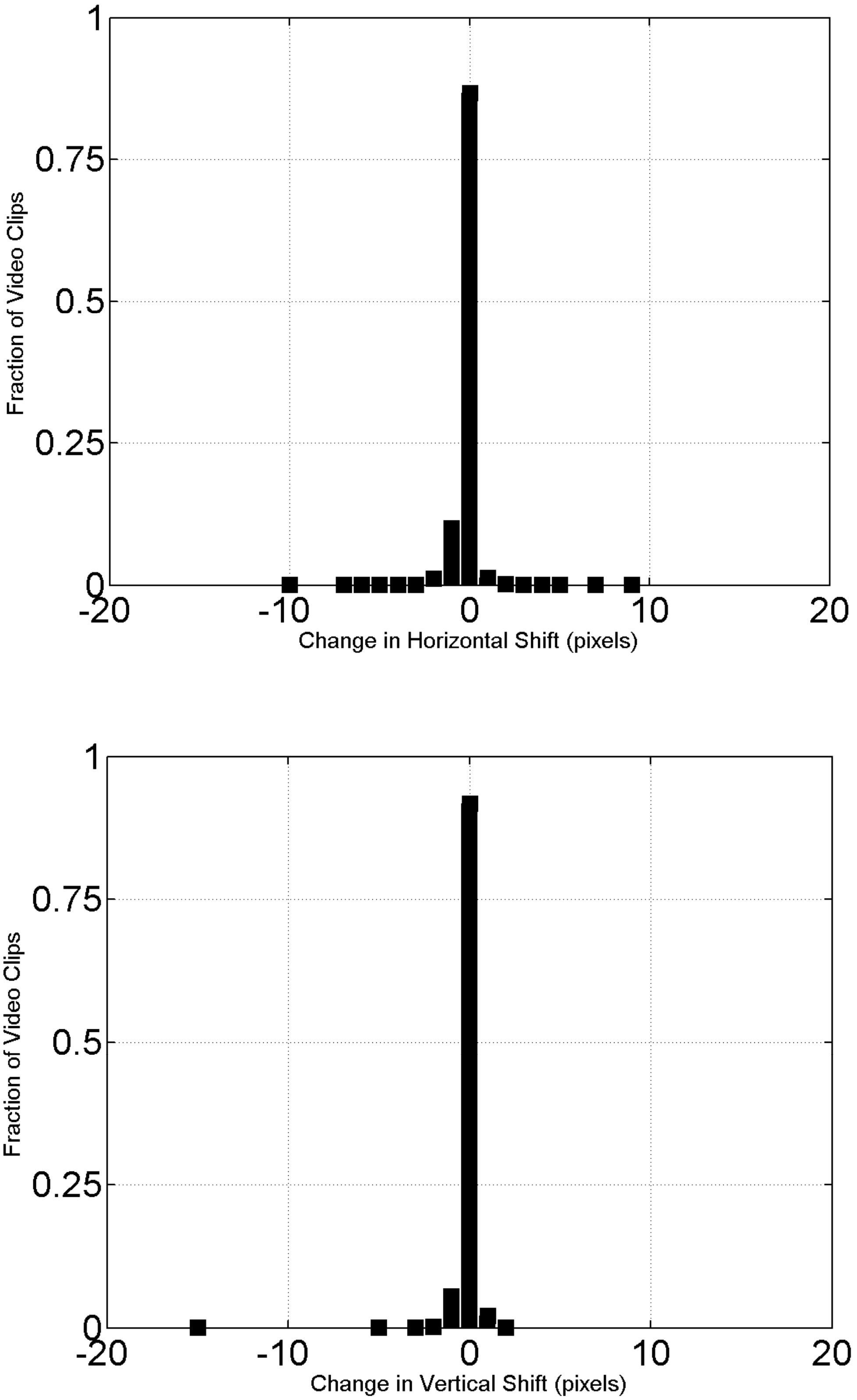


Figure A.5 – Difference between RR spatial shifts and the algorithm in clause 3.1.5 of [3]

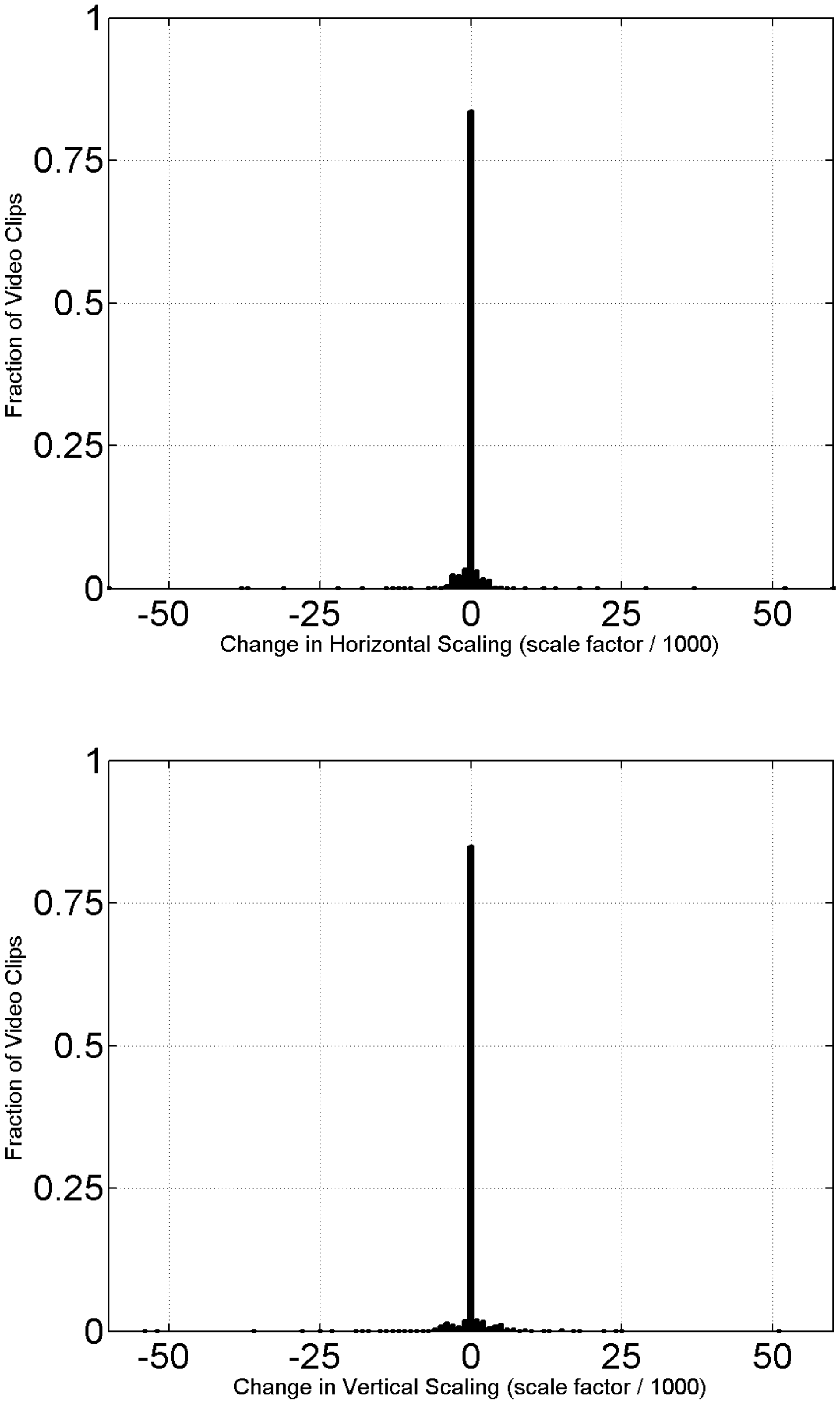


Figure A.6 – Difference between RR spatial scaling and the algorithm in [4]

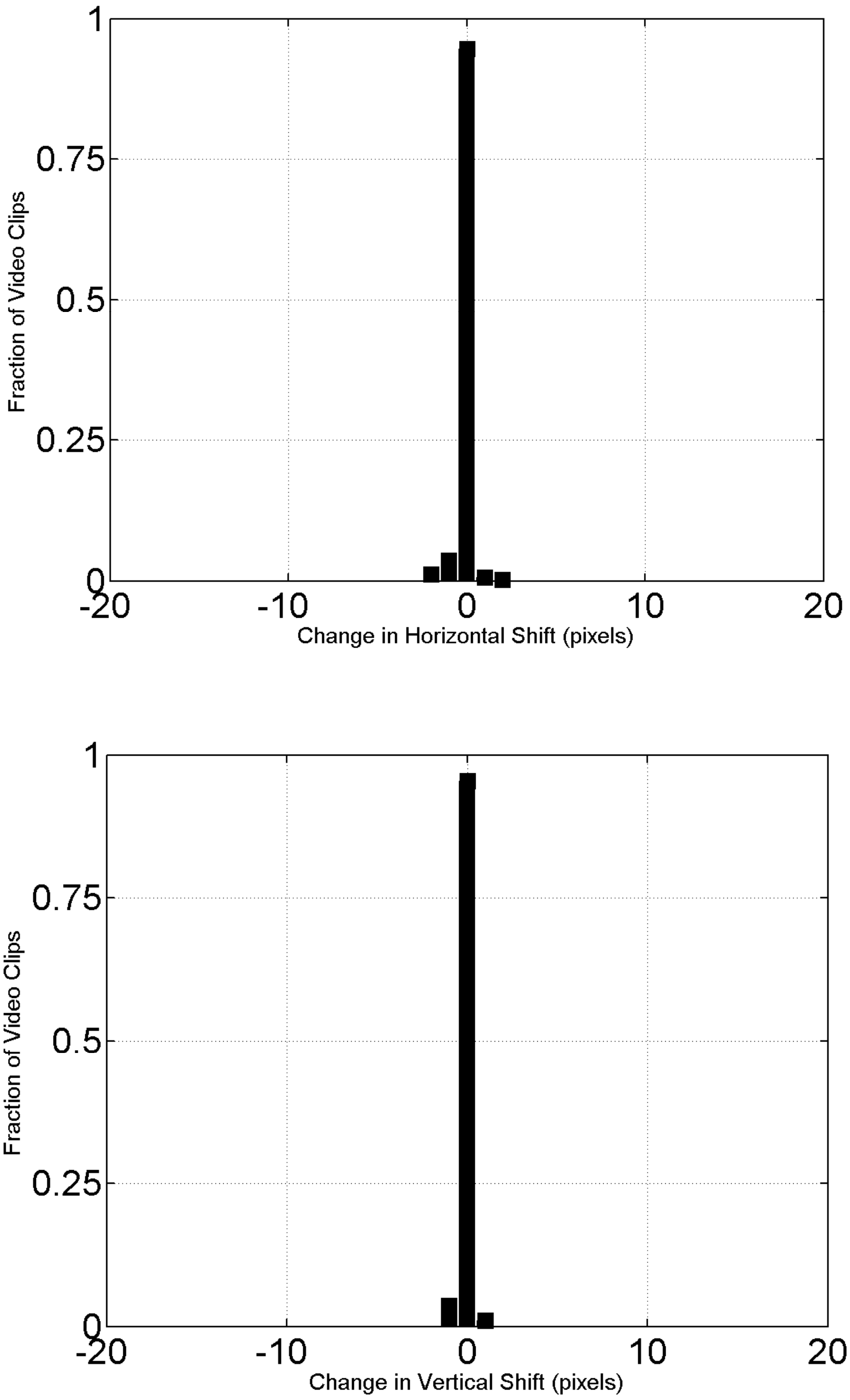
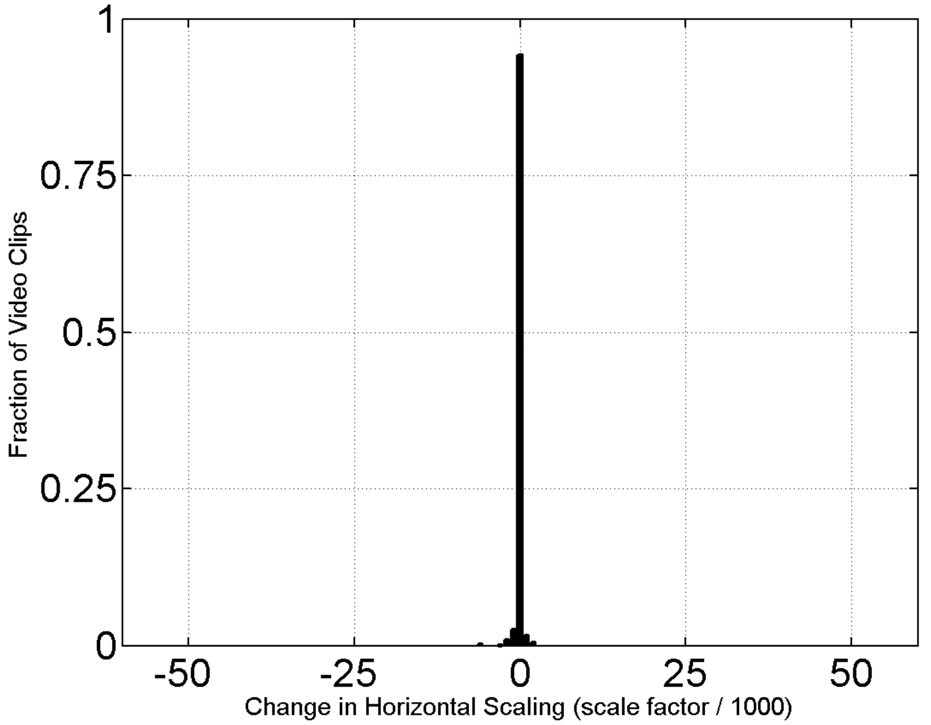


Figure A.7 – Difference between RR spatial shifts with median filtering (over scenes) and the algorithm in clause 3.1.5 of [3]



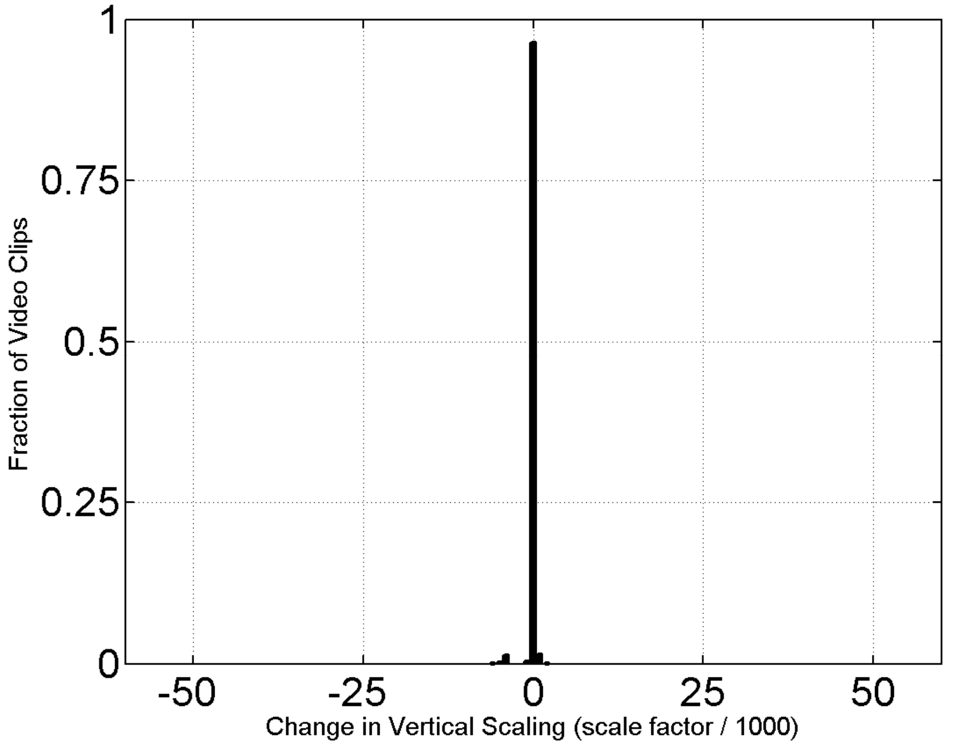


Figure A.8 – Difference between RR spatial scaling with median filtering (over scenes) and the algorithm in [4]

In general, the vast majority of the median filtered shifts agree within plus or minus one pixel; and almost all of the median filtered scaling factors agree within ±2 (i.e., ±0.2% scaling). Most of the non-zero differences between the algorithms come from video systems with four or fewer video scenes, or video systems that contained frequent digital transmission errors (e.g., significant transmission errors in every clip). Some of these differences may be indicative of truly ambiguous spatial registration. For example, if the video system blurs the video, performs field repetition (e.g., copies field one into field two), or utilizes lower resolution (e.g., 525-line video compressed to CIF and then up-converted and displayed at 525-line resolution), then spatial registration becomes ambiguous. Additionally, the reference algorithms used to define the "truth" data (i.e., the spatial registration algorithm in clause 3.1.5 of [3], and the spatial scaling algorithm in [4]) may produce some errors.

It is reasonable to ignore median filtered horizontal and vertical scaling factors (*vm* and *hm*) that are within ±2 of 0 (i.e., ±0.2% rescaling). Since most video systems do not perform this kind of minor rescaling, scaling factors within ±2 of 0 are more likely to be indicative of ambiguous spatial registration than true rescaling. Thus, for these cases we recommend replacing the rescaling results with no rescaling.

A.3.5 Observations and Conclusions

The reduced reference spatial registration algorithm presented here uses a set of low bandwidth RR features extracted from the original video sequence. One byte each is required to transmit each randomly chosen original pixel, and at most single precision will be required (i.e., 4-byte floating point) to transmit the original horizontal and vertical profiles. With this level of quantization, the algorithm requires an RR data transmission bit rate of approximately 27 kbit/s for 525-line video, and 30 kbit/s for 625-line video. Further bandwidth savings can be obtained through quantization (e.g., 16 bits per profile value).

This algorithm is suitable for measuring spatial shift and spatial scaling in a fully automated, in-service, real-time monitoring system or for an out-of-service environment, prior to performing video quality measurements. The video spatial scaling algorithm utilizes two types of features, pixels that track small local changes in scene content, and profiles that track overall trends. Accuracy is obtained by simultaneously estimating spatial shift and spatial scaling using the features extracted from multiple video frames. Results from multiple video sequences may be combined to further improve accuracy. This algorithm has been demonstrated to provide reliable estimates of spatial registration for a wide range of video scenes and systems.

A.4 Valid region algorithm

The valid region algorithm presented here uses a set of extremely low bandwidth RR features extracted from the original video sequence. Four integers are transmitted every time a valid region update is required (e.g., 16 bit/s to update the valid region estimate every four seconds).

A.4.1 Valid region when over-scan is present

NTSC (525-line) and PAL (625-line) video sampled according to Rec. 601 may have a border of pixels and lines that do not contain a valid picture. For instance, the original video from the camera may only fill a portion of the Rec. 601 frame. A digital video system that utilizes compression may further reduce the area of the picture in order to save transmission bits. If the non-transmitted pixels and lines occur in the over-scan area of the television picture, the typical end-user should not notice the missing lines and pixels. If these non-transmitted pixels and lines exceed the over-scan area, the viewer may notice a black border around the picture, since the receiving system will normally insert black into this non-transmitted picture area. Video systems (particularly those that perform low-pass filtering) may exhibit a ramping up from the black border to the picture area. These transitional effects most often occur at the left and right sides of the image but can also occur at the top or bottom. Occasionally, the processed video may also contain several lines of corrupted video at the top or bottom of the picture that may not be visible to the viewer (e.g., VHS tape recorders may corrupt several lines at the bottom of the picture in the over-scan area). The automated valid region algorithm presented here estimates the valid region of the original and processed video streams so that subsequent computations do not consider corrupted lines at the top and bottom of the Rec. 601 frame, black border pixels, or transitional effects where the black border meets the picture area.

A.4.1.1 Core valid region algorithm

This clause describes the core valid region algorithm that is applied to a single original or processed image. This algorithm requires three input arguments: an image, a maximum valid region, and the current valid region estimate.

Image. The core algorithm uses the Rec. 601 luminance image of a single video frame. When measuring the valid region of a *processed* video sequence, any spatial scaling and shift imposed by the video system must have been removed from the luminance image before applying the core algorithm (see clause A.3).

Maximum Valid Region. The core algorithm will not consider pixels and lines outside of a maximum valid video region. This provides a mechanism for the user to specify a maximum valid region that is smaller than the entire image area if *a priori* knowledge indicates that the sampled image has corrupted pixels or lines.

Current Valid Region. The current valid region is an estimate of the valid region and lies entirely within the maximum valid region. All pixels inside the current valid region are known to contain valid video; pixels outside the current valid region may or may not contain valid video content. Initially, the current valid region is set to the smallest possible area located at the exact centre of the image.

The core algorithm examines the area of video between the maximum valid region and the current valid region. If some of those pixels appear to contain valid video, the current valid region estimate is enlarged. The algorithm will now be described in detail for the left edge of the image.

1) Compute the mean of the left-most column of pixels in the maximum valid region. The left-most column of pixels will be denoted as column "J-1" and the mean will be represented by "M_{J-1}."

2) Take the mean of the next column of pixels, "M_J".

3) Column J is declared invalid video if it is black, ($M_J < 20$) or if the average pixel level of the mean value for successive columns indicates a ramp up from black border to valid picture ($M_J – 2 > M_{J-1}$). If either of these conditions is true, increment J and repeat steps 2) and 3). Otherwise, go to step 4).

4) If final column J is within the current valid region, then no new information has been obtained. Otherwise, update the current valid region with J as the left coordinate.

The algorithm for finding the top edge of the image is similar to that given above for the left edge. For the bottom and right edges, the algorithm is the same but the directions are reversed (e.g., J is decremented instead of incremented). The values produced for top, left, bottom, and right indicate the last valid pixel or line.

The stopping conditions identified in step 3) can be fooled by scene content. For example, an image that contains genuine black at the left side (i.e., black that is part of the scene) will cause the core algorithm to conclude that the left-most valid column of video is farther toward the middle of the image than it ought to be. For that reason, the core algorithm is applied to multiple images from a video sequence, thereby increasing the accuracy of the valid region estimate.

A.4.1.2 Applying the core valid region algorithm to a video sequence

Original video

The core algorithm is first applied to the original sequence of images. For NTSC video sampled according to Rec. 601, the recommended setting for the maximum valid region is top = 6, left = 6, bottom = 482, right = 714. For PAL video sampled according to Rec. 601, the recommended setting for the maximum valid region is top = 6, left = 16, bottom = 570, right = 704. The core algorithm is run on the first image in the video sequence and a uniform sub-sampling of images that follow. For example, the core algorithm could examine image numbers 0, 15, 30, 45, and so forth. When all selected images in the sequence have been examined, the current valid region will contain the largest valid area implied by any examined image in the video sequence. Pixels and lines between this final current valid region and the maximum valid region are considered to contain either black or a transitional ramp up from black.

The final valid region must contain an even number of lines and an even number of pixels. Any odd top or left coordinates are incremented by one. Then, if the region contains an odd number of lines, bottom is decremented; likewise, if the region contains an odd number of pixels (e.g., horizontally), right is decremented. This simplifies color processing for video sampled in accordance with Rec. 601, since the color channels are sub-sampled by 2 when compared to the luminance channel. Also, each interlaced field of video will contain the same number of video lines. This will ensure that spatial-temporal sub-regions (from which video quality features will be extracted) always contain valid video with equal contributions from both interlaced fields. The resulting valid region is returned as the original valid region.

Processed video

When computing the valid region of the processed video sequence, the maximum valid region setting for the core algorithm is first set equal to the corresponding original valid region found for that video scene. This maximum valid region is then reduced in size by any pixels and lines that are considered invalid due to spatial shift correction of the processed video frames. The core algorithm is then run on the first image in the processed video sequence and a uniform sub-sampling of the images that follow.

After the core algorithm has been applied to the processed video sequence, the valid region found by the core algorithm is reduced inward by a safety margin. The recommended safety margin discards one line off the top and bottom, and five pixels off the left and right. The large left and right safety margins ensure that any ramp up or down from black is excluded from the processed valid region.

The final processed valid region must contain an even number of lines and an even number of pixels. Any odd top or left coordinates are incremented by one. Then, if the region contains an odd number of lines, the bottom is decremented; likewise, if the region contains an odd number of pixels (e.g., horizontally), the right is decremented. The resulting valid region is returned as the processed valid region.

A.4.1.3 Comments on valid region algorithm

This automated valid region algorithm will work well to estimate the valid region of most scenes. Due to the nearly infinite possibilities for scene content, the algorithm described herein takes a conservative approach to estimation of the valid region. A manual examination of the valid region would quite likely result in a larger region. Conservative valid region estimates are more suitable for an automated video quality measurement system, because discarding a small amount of video will have little impact on the video quality estimate and in any case this video usually occurs in the over-scan part of the video. On the other hand, including corrupted video in the video quality calculations may have a large impact on the quality estimate.

This algorithm is not able to distinguish between corrupted pixels and lines at the edge of an image and true scene content. A rule of thumb is used instead, stating that such invalid video generally occurs at the extreme edges of the image. Specification of a conservative user-definable maximum valid video region (i.e., the starting point for the automated algorithm) provides a mechanism to exclude these possibly corrupt image edges from consideration.

A.4.2 Valid region when entire picture is displayed

CIF, SIF, QCIF, QSIF, and VGA are intended for computer display. These video sequences contain a valid picture all the way out to each edge. However, many of these sequences were originally shot using an NTSC, PAL or HDTV camera. As a result, these progressive video sequences occasionally have a border of pixels and lines that do not contain valid picture. This border is usually black, but a single line may sometimes contain values that are halfway between black and valid picture content (e.g., where the black border and valid picture values were averaged together). Often, this invalid border will occur on only one edge of the image. Digital video transmission systems usually do not further reduce the valid picture area, but may distort the sharp line from black to valid picture more than the rest of the picture. Viewers are used to ignoring black values around the edge of a picture, and will generally ignore this impairment. Thus, these border areas and related edge transitions should be excluded from video quality measurements. This case is further distinguished from the over-scan case because the number of pixels and lines can be quite small. Thus, extra care must be taken to preserve the maximum valid region possible. This clause presents a modified version of the valid region algorithm in clause A.4.1, where the modifications seek to preserve as much of the valid region as possible.

A.4.2.1 Modified Core Valid Region Algorithm

This clause describes the modified core valid region algorithm that is applied to a single original or processed image. This algorithm requires three input arguments: an image, a maximum valid region, and the current valid region estimate.

Image. The core algorithm uses the luminance image of a single video frame. When measuring the valid region of a *processed* video sequence, any spatial scaling and shift imposed by the video system must have been removed from the luminance image before applying the core algorithm (see clause A.3).

Maximum Valid Region. The core algorithm will not consider pixels and lines outside of a maximum valid video region. This provides a mechanism for the user to specify a maximum valid region that is smaller than the entire image area if *a priori* knowledge indicates that the sampled image has corrupted pixels or lines.

Current Valid Region. The current valid region is an estimate of the valid region and lies entirely within the maximum valid region. All pixels inside the current valid region are known to contain valid video; pixels outside the current valid region may or may not contain valid video content. Initially, the current valid region is set to contain 92% of the image horizontally and 92% of the image vertically. A border of 4% of the available pixels and lines is placed outside the current valid region at each image edge. Thus, the final valid region estimate will lie between this 4% point and the entire image.

The modified core algorithm examines the area of video between the edge of the picture and the current valid region. If some of those pixels appear to contain valid video, the current valid region estimate is enlarged. The algorithm will now be described in detail for the left edge of the image.

1) Compute the mean of the left-most column of pixels in the maximum valid region. The left-most column of pixels will be denoted as column "J" and the mean will be represented by "M_J".

2) Take the mean of the next column of pixels, "M_{J+1}".

3) Column J is declared invalid video if it is black ($M_J < 20$), or if the average pixel level of the mean value for successive columns indicates a steep ramp[5] up from black border to

[5] The difference in the slope of the ramp between clauses A.4.1 and A.4.2 is intentional. Design differences between these types of video codecs influence this constant.

valid picture ($M_J + 20 < M_{J+1}$). If either of these conditions is true, increment J and repeat steps 2) and 3). Otherwise, go to step 4).

4) If final column J is within the current valid region, then no new information has been obtained. Otherwise, update the current valid region with J as the left coordinate.

The algorithm for finding the top edge of the image is similar to that given above for the left edge. For the bottom and right edges, the algorithm is the same but the directions are reversed (e.g., J is decremented instead of incremented). The values produced for top, left, bottom, and right indicate the last valid pixel or line.

The stopping conditions identified in step 3) can be fooled by scene content. For example, an image that contains genuine black at the left side (i.e., black that is part of the scene) will cause the core algorithm to conclude that the left-most valid column of video is farther toward the middle of the image than it ought to be. For that reason, the core algorithm is applied to multiple images from a video sequence, thereby increasing the accuracy of the valid region estimate.

A.4.2.2 Applying the modified core valid region algorithm to a video sequence

Original video

The modified core algorithm is first applied to the original sequence of images. The core algorithm is run on the first image in the video sequence and a uniform sub-sampling of the images that follow. For example, the core algorithm could examine image numbers 0, 15, 30, 45, and so forth. When all selected images in the sequence have been examined, the current valid region will contain the largest valid area implied by any examined image in the video sequence. Pixels and lines between this final current valid region and the edge of the image are considered to contain either black or a transitional ramp up from black. The resulting valid region is returned as the original valid region.

Processed video

When computing the valid region of the processed video sequence, the maximum valid region setting for the core algorithm is first set equal to the corresponding original valid region found for that scene. This maximum valid region is then reduced in size by any pixels and lines that are considered invalid due to spatial scaling and shift correction of the processed video frames. The modified core algorithm is then run on the first image in the processed video sequence, and a uniform sub-sampling of the images that follow. The resulting valid region is returned as the processed valid region.

A.5 Luminance gain and offset algorithm

The luminance gain and offset algorithm presented here uses a set of low bandwidth RR features extracted from the original video sequence. The algorithm presented here may be implemented as either a downstream RR algorithm or an upstream RR algorithm, since identical information is required from the processed and original video sequences. The luminance gain and offset algorithm requires an RR data transmission bit rate of 0.5 to 1.4 kbit/s (depending upon image size and feature quantization accuracy).

This luminance gain and offset algorithm replaces an algorithm presented in an earlier version of this report (i.e., NTIA Report TR-06-433). That algorithm reused data from the spatial registration algorithm, and so could be computed without requiring additional RR data transmission bandwidth. Upon further analysis, the earlier algorithm exhibited poor behavior for some scene content, including biased estimation errors.

A.5.1 Preconditions

For the luminance gain and offset algorithm to operate reliably, the original and processed video sequences must be temporally aligned and spatially registered. Additionally, the valid video region must be calculated.

The luminance gain and offset algorithm does not perform *any* temporal search of its own. Temporal registration using the estimated average delay is presumed to be sufficiently accurate. This approach was found to yield accurate results for processed video sequences exhibiting constant or variable frame rate encoding. The training data used to develop the luminance gain and offset algorithm included transmission errors, and these errors produced pausing with loss, pausing without loss, and corrupted images. The training data also included a wide range of frame rates, including variable frame rate encoding that occasionally was slower than 1 fps.

Because this algorithm does not do a temporal search, it is able to treat progressive and interlaced video sequences identically. Complications resulting from interlaced field reframing only arise when field streams are slid past each other. The elimination of temporal registration search from the spatial registration algorithm eliminates the need to treat interlaced video differently from progressive video.

A.5.2 Algorithm overview

The luminance gain and level offset estimation examines one pair of original and processed frames every second. The processed video must be corrected for spatial scaling and shift, as appropriate. The border of pixels and lines outside the valid region (see clause A.4) should be eliminated from both original and processed frames.

Each video frame is divided spatially into blocks. Table A.4 gives reasonable default values of these constants for common image sizes. The luminance values within each block are averaged, to create sub-sampled versions of the original and processed video sequences. The sub-sampled values for all original frames examined (one per second) are shaped into a single one-dimensional array, named $\underline{O}$. The sub-sampled values for all processed frames examined (one per second) are shaped into a single one-dimensional array, named $\underline{P}$. Notice that respective elements of vectors $\underline{O}$ and $\underline{P}$ represent the same location in space (horizontally and vertically) and time (frame number) within the original and processed video sequences, respectively.

Table A.4 – Default block size for luminance gain and level offset

NTSC, PAL, VGA	CIF, SIF	QCIF, QSIF
46	22	10

For increased robustness of the gain/offset least squares fitting algorithm, it is desirable to prefilter the $\underline{O}$ and $\underline{P}$ vectors to eliminate those blocks that contain a wide spread of pixel values and those blocks that have been clipped. Blocks that contain a wide spread of pixel values have luminance transitions and these transitions are more affected by spatial miss-alignments and video distortions. Blocks that have been clipped have lost useful information for estimating gain and level offset. First identify those original blocks with the greatest spread in pixel values by calculating the standard deviation of each block (call this vector Os) and then eliminate 50% of the blocks (from both the $\underline{O}$ and the corresponding $\underline{P}$ vector) that have the greatest standard deviation. Next, examine vectors $\underline{O}$ and $\underline{P}$ and eliminate those block pairs that are less than 2 or greater than 253 in either the $\underline{O}$ or $\underline{P}$ vector. After elimination of these blocks, if the maximum of P minus the minimum of P is less than 10 do not attempt to computer gain/offset as the range of pixel values is insufficient.

Calibration involves computing the gain (g) and level offset (l) according to the following model:

$$\underline{P} = g\underline{O} + l \tag{40}$$

There are only two unknowns (i.e., g and l) but Z equations (one equation for each block pair). Therefore, we must solve the over-determined system of linear equations given by:

$$\underline{P} \approx \underline{\hat{P}} = A\begin{bmatrix} l \\ g \end{bmatrix} \tag{41}$$

where A is a Z x 2 matrix given by

$$A_{Z\text{x}2} = \begin{bmatrix} \underline{1} & \underline{O} \end{bmatrix} \tag{42}$$

and $\underline{1}$ is a Z-element column vector of '1s' given by

$$\underline{1}_{Z\text{x}1} = \begin{bmatrix} 1_1 \\ \cdot \\ \cdot \\ \cdot \\ 1_Z \end{bmatrix} \tag{43}$$

$\hat{P}$ is the estimate of the processed samples if the gain and level offset correction were applied to the original samples. The least squares solution to this over-determined problem is given by

$$\begin{bmatrix} l \\ g \end{bmatrix} = \left(A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}}P \tag{44}$$

where the superscript "T" denotes matrix transpose and the superscript "–1" denotes matrix inverse. Problems with using equation (44) directly include using processed video samples that are highly distorted from the original video samples, and using processed video samples that are not correctly registered in space and time to the original video samples (e.g., the processed video may contain frame repeats and variable video delay). Thus, an iterative least squares solution with a cost function is used to help minimize the weight of these outliers in the fit.

The following iterative algorithm is applied to the original and processed samples:

1) Use the normal least squares solution, equation (44), to generate the initial estimate of the level offset and gain.

2) Generate an error vector ($\underline{E}$) that is equal to the absolute value of the difference between the true processed samples and the fitted processed samples:

$$\underline{E} = \left|\underline{P} - \underline{\hat{P}}\right| \tag{45}$$

3) Generate a cost vector ($\underline{C}$) that is the element-by-element reciprocal of the error vector ($\underline{E}$) plus a small epsilon (ε):

$$\underline{C} = \frac{1}{\underline{E} + \varepsilon} \tag{46}$$

The ε prevents division by zero and sets the relative weight of a point that is on the fitted line versus the weight of a point that is off the fitted line. An ε of 1.0 is recommended.

4) Normalize the cost vector $\underline{C}$ for unity norm (i.e., each element of $\underline{C}$ is divided by the square root of the sum of the squares of all the elements of $\underline{C}$).

5) Generate the cost vector $\underline{C}^2$ that is the element-by-element square of the cost vector $\underline{C}$ from step 4.

6) Generate an N x N diagonal cost matrix (C^2) that contains the cost vector's elements ($\underline{C}^2$) arranged on the diagonal, with zeros everywhere else.

7) Using the diagonal cost matrix (C^2) from step 6, perform cost-weighted least squares fitting to determine the next estimate of the level offset and gain:

$$\begin{bmatrix} l \\ g \end{bmatrix} = \left(A^{\mathrm{T}} C^2 A\right)^{-1} A^{\mathrm{T}} C^2 \underline{P} \tag{47}$$

8) Repeat steps 2 through 7 until the level offset and gain estimates converge to four decimal places.

To remove gain and level offset from the processed *Y* image plane, the following formula is applied to each processed pixel:

$$\text{New } Y(i,j,t) = [Y(i,j,t) - l]/g \tag{48}$$

Similar to that found for spatial registration in clause A.3.4, we have found that median filtering across scenes produces more robust estimates for luminance gain and offset.

The above algorithm for estimating luminance (Y) gain and offset may also be applied to the chrominance channels (Cb and Cr).

A.6 Combining algorithms and applying corrections

To fully calibrate a processed video sequence, we suggest performing the calibration in the following order. The calibration information from earlier steps is used in subsequent steps. First, estimate the temporal registration of the uncalibrated processed video sequence using the algorithm in clause A.2 and apply the appropriate temporal correction. When examining interlaced video, temporally register to field accuracy. Second, estimate the spatial scaling and shift using the algorithm in clause A.3. Correcting for vertical spatial shifts may involve reframing of the processed video. Third, estimate the valid region using the algorithms in clause A.4. Fourth, estimate the luminance gain and level offset using the algorithm in clause A.5. Fifth, reapply the temporal registration algorithm in clause A.2 using the calibrated processed video clip to obtain a slightly improved temporal registration estimate. Temporally register to frame accuracy for both interlaced and progressive video sequences. This fifth step is optional for progressive video sequences that do not have spatial registration issues (i.e., zero spatial shift and no spatial scaling).

If spatial scaling, spatial shift, luminance gain, and level offset estimates are available for other processed video sequences that have passed through the same video system (i.e., all video sequences can be considered to have the same calibration numbers, except for temporal registration), then calibration results can be filtered across scenes to achieve increased accuracy. We have found that median filtering across scenes produces robust estimates for spatial scaling, spatial shift, luminance gain, and level offset. In this case, the fourth and fifth steps given above can result in improved accuracy by median filtering the values for spatial scaling, spatial shift, luminance gain, and level offset.

Taken together, these four algorithms accurately estimate video calibration values in an RR measurement environment. The required bandwidth of the NTIA/ITS RRCAL algorithm is 25.028 bit/s for 525-line/NTSC video, and 25.856 bit/s for 625-line/PAL when operating on Recommendation ITU-R BT.601 video. This bandwidth should fit comfortably within commonly available IP bandwidths. The breakdown of bandwidth per algorithm is as follows:

- Temporal registration: 2.160 bit/s NTSC and 1.800 bit/s PAL, assuming quantization to 12 bits – this algorithm typically runs twice, so the stated bit rate must be doubled.
- Spatial shift/spatial scaling: 18.368 bit/s NTSC and 19.428 bit/s PAL, assuming quantization of 8 bits per pixel and 10 bits for other feature values.
- Gain and offset (Y, Cb, Cr): 2.276 bit/s NTSC and 2.764 bit/s PAL, assuming quantization to 10 bits, with 1 bit per $\underline{Os}$ block (keep or discard).
- Valid region: 64 bit/s maximum.

Image sizes (e.g., CIF, QCIF) and frame rates (e.g., 15 fps) different from Recommendation ITU-R BT.601 will require different bandwidths. For example, the requirements for CIF and QCIF with the above quantizer assumptions are as follows:

- Temporal registration bandwidth requirements decrease proportionately with frame rate, where the bit rate required for progressive video is half that required for interlace. 15 fps progressive video requires 540 bit/s.
- Spatial shift/spatial scaling bandwidth requirements decrease with image size. CIF bandwidth requirements are 10.496 bit/s, and QCIF bandwidth requirements are 5.248 bit/s.
- Gain and offset (Y, Cb, & Cr) bandwidth requirements increase slightly with decreasing image size (due to the smaller block sizes that are used). CIF bandwidth requirements are 3.328 bit/s and QCIF bandwidth requirements are 3.808 bit/s.
- Valid region: 64 bit/s maximum.

A.7 References

[1] Recommendation ITU-T J.143 (2000), *User requirements for objective perceptual video quality measurements in digital cable television*.

[2] Recommendation ITU-R BT.601 (1994), *Encoding parameters of digital television for studios*.

[3] Wolf, S. and Pinson, M. (2002), *Video quality measurement techniques*, NTIA Report 02-392. Jun. Available at <www.its.bldrdoc.gov/n3/video/documents.htm>.

[4] Pinson, M. and Wolf, S. (2005), *Video scaling estimation technique*, NTIA Technical Memorandum TM-05-417, Jan. Available at <www.its.bldrdoc.gov/n3/video/documents.htm>.

Annex B

Yonsei University

A calibration method for constant misalignment of spatial and temporal domains with constant gain and offset

(This annex forms an integral part of this Recommendation)

B.1 Introduction

Figure B.1 illustrates the overall procedure. First, spatial/temporal registration with the full search ranges is performed. Then, gain and offset are estimated. With the estimated gain and offset values, spatial/temporal registration is repeated with reduced search ranges. If the two registration results are identical, the calibration method outputs the results. Otherwise, the procedure is iterated.

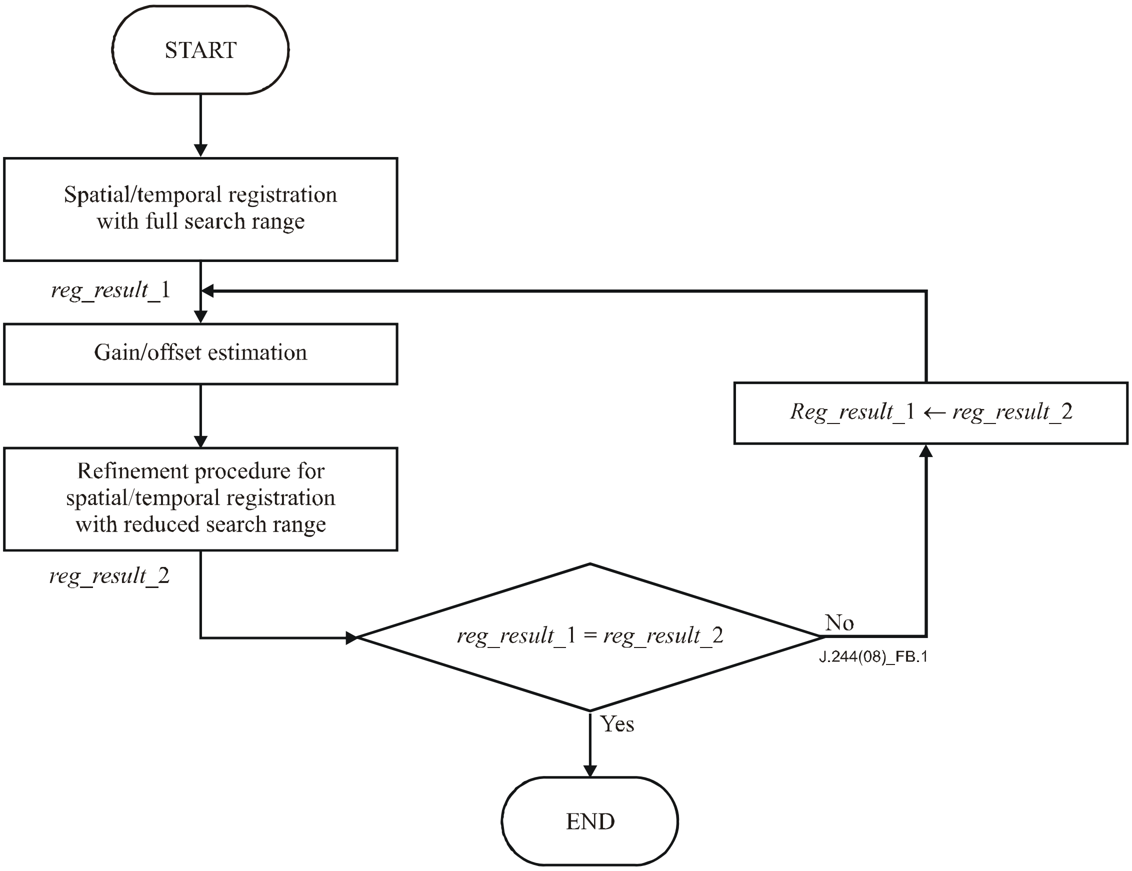

START
Spatial/temporal registration with full search range
reg_result_1
Gain/offset estimation
Refinement procedure for spatial/temporal registration with reduced search range
reg_result_2
reg_result_1 = reg_result_2
No
Yes
Reg_result_1 ← reg_result_2
END
J.244(08)_FB.1

Figure B.1 – Flowchart of the calibration method

B.2 Spatial and temporal registration

First, the calibration method randomly selects pixels from source video sequences within the effective area (Figure B.2) for ITU-R 601. Table B.1 shows the number of pixels selected from each field. Then, the entire pixels are used to find the best match with processed video sequences. In other words, using a full search method, the spatial/temporal registration routine of the calibration method finds the best spatial and temporal shifts which provide the minimum mean squared error. It is noted that the locations of the randomly selected pixels of the source video sequence can be transmitted by just sending the random seed which is used in the random number generator. Each pixel is represented by 8 bits.

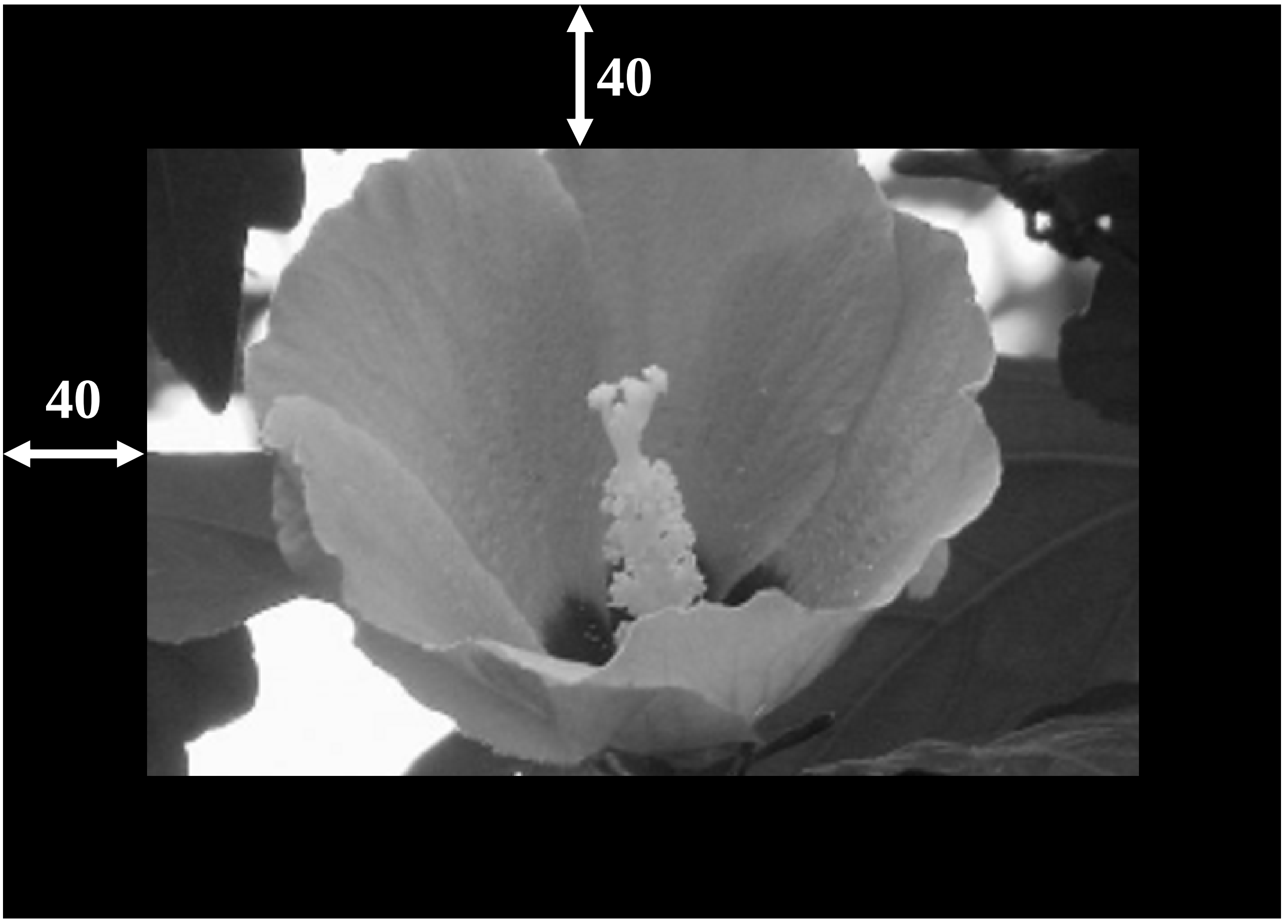


Figure B.2 – Effective area

Table B.1 – Number of pixels and number of gain/offset features per field

Models	Number of pixels	Number of gain/ offset features
Unlimited	50	900
56k	21	41
10k	8	5
5k	4	2
1k	2	
0.5k	1	

B.3 Gain and offset estimation

Gain and offset can be best estimated from homogeneous regions. In order to find such homogeneous regions, the calibration method applies the Sobel operator to each field and selects the regions which contain lowest numbers of edge pixels (Figure B.3). For instance, to find edge pixels, the modified successive gradient operation in [ITU-T J.144] can be used with a threshold value of 260. The average values of these regions are used to estimate gain and offset using the least square estimation. These average values will be called as "gain/offset features". Table B.1 shows the number of these gain/offset features selected from each field. For 1k and 0.5k, pixel values will be gain/offset features. The locations of the regions should be encoded and transmitted. It is noted that at this point the source and processed video sequences are aligned in the spatial and temporal directions (Figure B.1).

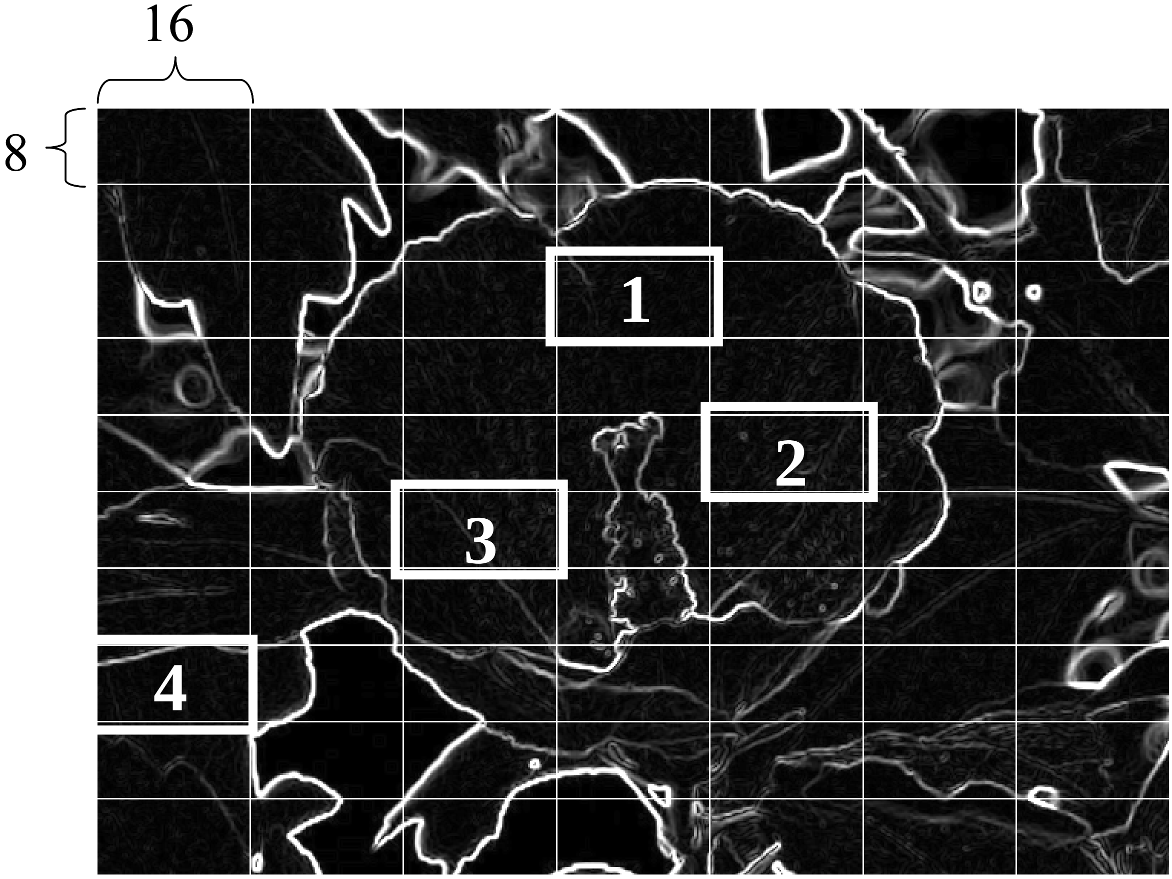


Figure B.3 – Selection of homogeneous regions

In order to find the gain and offset values, the gain and offset values are estimated using the gain/offset features of a group of six fields (Figure B.4). Assuming that there are N gain/offset features, the gain and offset values are estimated as follows:

$$y_i = ax_i + b \qquad i=1,\dots,N$$

where a is the gain value, b the offset value, x_i a gain/offset feature computed from the source video sequence and y_i the corresponding gain/offset feature of the processed video sequence. For example, if there are 480 fields, this procedure will produce 80 gain and offset features. The least square estimation method is used. This operation will produce a series of gain and offset values. Then, median filtering is applied to these values and finally the average values after median filtering are used as the estimated gain and offset values.

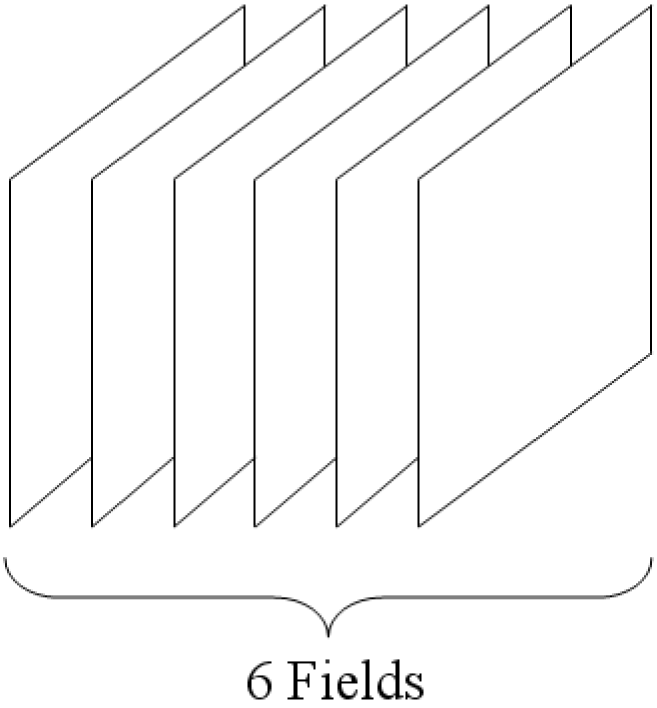


Figure B.4 – Estimating gain and offset values using the gain/offset features of a group of six fields

B.4 Refinement

Using the estimated gain and offset values, spatial and temporal registrations are repeated with reduced search ranges (spatial: +/– 3 pixels, temporal: +/1 one second). If the spatial and temporal shifts remain unchanged, the procedure stops. Otherwise, the procedure is repeated (Figure B.1).

Appendix I

Example Validation Testplan for Calibration Methods

(This appendix does not form an integral part of this Recommendation)

I.1 Introduction

In the last ITU-T SG 9 meeting in May 2005, Questions 2/9 and 14/9 were revised to allow development of new Recommendations for calibration methods for video sequences. In this appendix, the validation testplan for such methods is described.

There are a number of factors that can cause spatial and temporal misalignment, gain and offset. Encoders may cause spatial cropping, gain and offset. Transmission errors in digital communications may cause temporal shift. Analog paths in the transmission channel may cause gain, offset, temporal misalignment, spatial displacement. Combination of all these error sources may produce a prohibitively large number of test conditions, which may be simultaneously tested. Therefore, the recommendations for calibration will be developed in several phases.

As an initial effort, the test conditions described next will be evaluated. Based on the outcome of the evaluation tests, some models will be included in the Recommendations.

I.2 Example Calibration test plan (no variable temporal misalignment)

In the first calibration validation, models will be tested for spatial and temporal misalignment. Variable temporal misalignment and variable frame rates observed in multimedia transmission will not be tested. The bandwidth for reduced-reference models will be given in Table I.1, where unlimited bandwidth indicates that whole source video sequences will be available for calibrations.

Table I.1 – Allowed bandwidth for the Phase 1 Calibration Test

	Available bandwidth for registration					
ITU-R 601 (interlaced)	Unlimited	56 kbit/s	10 kbit/s	5 kbit/s	1 kbit/s	0.5 kbit/s
VGA (progressive)	Unlimited	56 kbit/s	10 kbit/s	5 kbit/s	1 kbit/s	0.5 kbit/s
CIF (progressive)	Unlimited	56 kbit/s	10 kbit/s	5 kbit/s	1 kbit/s	0.5 kbit/s
QCIF (progressive)	Unlimited	56 kbit/s	10 kbit/s	5 kbit/s	1 kbit/s	0.5 kbit/s
HD (interlaced)	Unlimited	56 kbit/s	10 kbit/s	5 kbit/s	1 kbit/s	0.5 kbit/s
HD (progressive)	Unlimited	56 kbit/s	10 kbit/s	5 kbit/s	1 kbit/s	0.5 kbit/s

Table I.2 shows the test conditions for the Phase 1 Calibration Test. Processed video sequences will be generated using a number of available source video sequences. For instance VQEG Phase I data can be used. If VGA, CIF and QCIF videos are to be made from VQEG Phase I data, the resizing and deinterlacing should be done in a proper manner.

Table I.2 – Test conditions for the Phase 1 Calibration Test

Source video length	8 seconds
Maximum spatial shift (vertical and horizontal)	±20 pixels/frame lines
Maximum temporal shift	±1 second
Gain (Y-component)	±10%
Offset (Y-component)	±10 grey levels

Processed video sequences will not be generated using analog parts since correct answers are not available for such PVSs. Similarly, processed video sequences will have integer spatial shifts and frame temporal shift. No subpixel misalignment will be allowed and no field temporal shift will be allowed since correct answers are not available for such PVSs.

It may be advantageous to test the various aspects of the calibration algorithms one at a time (e.g., gain/offset, spatial shift, temporal shift) in addition to testing the composite algorithm that includes all possible calibration problems. The reason is that a given model might have, say, the best gain/offset algorithm, but have a poor spatial shift algorithm. Table I.3 shows the possible combinations and the validation process will test all the combinations.

Table I.3 – Test condition combinations

	Gain/Offset	Spatial Shifts	Temporal shifts
Case 1	Yes	No	No
Case 2	Yes	No	No
Case 3	Yes	No	No
Case 4	Yes	Yes	No
Case 5	Yes	No	Yes
Case 6	No	Yes	Yes
Case 7	Yes	Yes	Yes

I.3 Validation process

The validation will be cross-checked by proponents. If there are any voluntary independent laboratories, the methods should be evaluated by the voluntary independent laboratories. The proponents must send their executable file to the voluntary independent laboratories and the voluntary independent laboratories should report the results. The laboratory which generates test video sequences must ensure that the exact amounts of spatial shift, temporal shift, gain and offset are known.

I.4 Generation of Processed Video Sequences (PVSs)

NOTE – In each condition, the exact answer must be known for the validation test.

- Spatial shift resolution: integer pixel shift; range: ±20 pixels/frame lines.
- Temporal shift resolution: integer field shift (integer frame shift for progressive cases); range: ±1 second.
- Gain and offset: Luminance only (±10% Gain and ±10 grey levels for offset case). No overflow or underflow (e.g., clipping) is allowed. (Resolution: floating point 4 significant digits).
- Wide range of source video (spatial and temporal detail) and quality levels at various bit rates.

- Fixed frame rate only for first test.
- It is recommended that the distribution of calibration errors in the space of gain, offset, x-shift, y-shift and temporal should have a uniform distribution.
- File format for PVSs and SRCs: uncompressed UYUV AVI.
- At least five difference sources should be used for each test condition.

I.5 Evaluation metrics

- Spatial Accuracy.
- Temporal Accuracy.
- Gain and offset Accuracy.
- The report should include at least the following statistics:
  - E_x: error in x-shifts (pixels)
  - E_y: error in y-shifts (pixels)
  - E_t: error in temporal direction (frame, one file=0.5)
  - E_g: error in gain $E_g = \frac{|G_{true} - G_{estimate}|}{G_{true}}$
  - E_o: error in offset (pixel value)
    1) percentiles of correct results (no errors in any aspect)
    2) percentiles of results with $E_x + E_y \leq 1$
    3) percentiles of results with $E_x + E_y + E_t \leq 1$
    4) percentiles of results with $E_x + E_y + E_t \leq 2$

I.6 Additional information

In reduced-reference video quality models, the processing time is an important issue and a fast registration method is desirable for real-time implementation. Although it is difficult to measure the processing time of a given model since the processing time depends heavily on the programmer's skill and hardware, it is possible that a proponent can estimate the number of required operations (comparisons, additions, multiplications, etc.). Thus, it is recommended that the proponents provide information on the complexity of their models in terms of required operations.

I.7 Future test plans (variable temporal misalignment for multimedia applications)

The next phases of registration methods will be tested on a multimedia environment, where transmission errors introduce variable temporal misalignment. Variable frame rates are also allowed. Furthermore, the duration of PVS can be reduced to a single frame. Table I.4 summarizes the future test conditions.

Table I.4 – Test conditions for future calibration tests

Duration of PVS	4, 2, 1, 0.5 sec (down to a single frame)
Variable frame rate	Allowed
Variable temporal misalignment	Allowed
Constant spatial scaling (i.e., stretching or shrinking of the video picture)	Plus or minus 6%

Appendix II

Test Plan 1 (Rec. 601 only – Jun 2, 2007)

(This appendix does not form an integral part of this Recommendation)

II.1 Calibration test plan 1 (no variable temporal misalignment)

In the first calibration validation, models will be tested for spatial and temporal misalignment. Variable temporal misalignment and variable frame rates observed in multimedia transmission will not be tested. The bandwidth for reduced-reference models will be given in Table II.1, where unlimited bandwidth indicates that whole source video sequences will be available for calibrations.

Table II.2 shows the test conditions for the Phase 1 Calibration Test. Processed video sequences will be generated using a number of available source video sequences. For instance VQEG Phase I data can be used but not exclusively.

Processed video sequences will not be generated using analog in any portion of the system since correct answers are not available for such PVSs. Similarly, processed video sequences will have integer spatial shifts and field or frame temporal shifts. Subpixel misalignments are not allowed. This should be verified by automatic methods as well as by visual inspection.

Table II.1 – Allowed bandwidth for the Phase 1 Calibration Test

	Available bandwidth for registration					
ITU-R 601 (interlaced)	Unlimited	56 kbit/s	10 kbit/s	5 kbit/s	1 kbit/s	0.5 kbit/s

Table II.2 – Test conditions for the Phase 1 Calibration Test

Source video length	8 seconds
Maximum spatial shift (vertical and horizontal)	±20 pixels/frame lines
Maximum temporal shift	±1 second
Gain (Y-component)	±10%
Offset (Y-component)	±10 grey levels

II.2 Validation process

The validation will be cross-checked by proponents. If there are any voluntary independent laboratories, the methods should be evaluated by the voluntary independent laboratories. The proponents must send their executable file to the voluntary independent laboratories and the voluntary independent laboratories should report the results. The laboratory which generates test video sequences must ensure that the exact amounts of spatial shift, temporal shift, gain and offset are known.

II.3 Generation of Processed Video Sequences (PVSs)

NOTE – In each condition, the exact answer must be known for the validation test. Each proponent will produce de-calibrated test conditions and keep the truth information secret from the other proponents. After all results are available, comparison with all of the truth data will be made to determine the validation results.

Source video will have markers added before processing through various coding and transmission systems. This will allow for exact alignment and calibration prior to the de-calibration step.

Decalibration will be done using software with parameter files used as input. The parameter files will specify spatial shift, temporal shift, gain and offset within the limits specified below:

- Spatial shift resolution: integer pixel shift; range: ±20 pixels.
- Temporal shift resolution: integer field shift; range: ±2 frames.
- Gain and offset: Luminance only (±10% Gain and ±10 grey levels for offset case) (Resolution: floating point 4 significant digits).
- Wide range of source video (spatial and temporal detail) and quality levels at various bit rates.
- Fixed frame rate only for this test.

II.4 Evaluation metrics

- Spatial Accuracy.
- Temporal Accuracy.
- Gain and offset Accuracy.
- The report should include at least the following statistics:
  - E_x : error in x-shifts (pixels)
  - E_y : error in y-shifts (pixels)
  - E_t : error in temporal direction (frame, one file = 0.5)
  - E_g : error in gain $E_g = \frac{|G_{true} - G_{estimate}|}{G_{true}}$
  - E_o : error in offset (pixel value)
    1) percentiles of correct results (no errors in any aspect)
    2) percentiles of results with $E_x + E_y \le 1$
    3) percentiles of results with $E_x + E_y + E_t \le 1$
    4) percentiles of results with $E_x + E_y + E_t \le 2$

Appendix III

Evaluation Metrics

(This appendix does not form an integral part of this Recommendation)

III.1 Introduction

A total of 1190 test video sequences (525 video format) are generated for various test conditions using difference source video sequences and HRCs. The same number of test video sequences is generated for the 625 video format.

The following five error metrics are computed in absolute values:

- E_x : error in x-shifts (pixels)
- E_y : error in y-shifts (pixels)
- E_t : error in temporal direction (frame)
- E_g : error in gain ($E_g = \frac{|G_{true} - G_{estimate}|}{G_{true}}$)
- E_o : error in offset (pixel value)

Then, the following percentiles are computed as evaluation metrics:

- Percentiles of results with $E_g + E_o + E_x + E_y + E_t < 0.0001$
- Percentiles of results with $E_x + E_y \leq 0$
- Percentiles of results with $E_x + E_y \leq 1$
- Percentiles of results with $E_x + E_y + E_t \leq 0$
- Percentiles of results with $E_x + E_y + E_t \leq 1$
- Percentiles of results with $E_x + E_y + E_t \leq 2$
- Percentiles of results with $E_t \leq 0$
- Percentiles of results with $E_t \leq 1$
- Percentiles of results with $E_t \leq 2$
- Percentiles of results with $E_g \leq 0.01$
- Percentiles of results with $E_g \leq 0.02$
- Percentiles of results with $E_o \leq 2.5$
- Percentiles of results with $E_o \leq 5.0$

III.2 Experimental results of 1190 PVSs for each video format

Tables III.1-4 show the mean and standard deviation values of the five errors and error statistics for the 525 video format. Tables III.5-8 show the mean and standard deviation values of the five errors and error statistics for the 625 video format.

Table III.1 – Mean values of the five errors (525 format, 1190 PVSs)

	Mean (Ex)	Mean (Ey)	Mean (Et)	Mean (Eg)	Mean (Eo)
NTIA 25k	0	0	–0.015	0.005	1.111
Yonsei unlimited	0	–0.00084	0.006723	0.005649	0.621204
Yonsei 56k	0	0.007563	0	0.005799	0.62936
Yonsei 10k	–0.00168	–0.00168	0.014286	0.006	0.624177
Yonsei 5k	0.00084	0.007563	0.002521	0.006184	0.590344
Yonsei 1k	0.002521	–0.00252	0.021008	0.014237	–0.56031
Yonsei 0.5k	–0.00924	0.010084	0.058824	0.01666	–0.68251

Table III.2 – Standard Deviation values of the five errors (525 format, 1190 PVSs)

	std (Ex)	std (Ey)	std (Et)	std (Eg)	std (Eo)
NTIA 25k	0	0	0.141	0.009	1.992
Yonsei unlimited	0	0.028989	0.091461	0.008178	1.301323
Yonsei 56k	0.058001	0.19002	0.142074	0.008753	1.451585
Yonsei 10k	0.057977	0.082009	0.132127	0.009016	1.542557
Yonsei 5k	0.076724	0.210993	0.282653	0.009271	1.680862
Yonsei 1k	0.112291	0.207091	0.527203	0.01491	2.098649
Yonsei 0.5k	0.161204	0.231786	1.941942	0.017672	2.431145

Table III.3 – Error statistics (525 format, 1190 PVSs)

Percentile	Ex+Ey≤0	Ex+Ey≤1	Ex+Ey+Et=0	Ex+Ey+Et≤1	Ex+Ey+Et≤2
NTIA 25k	100	100	98.739	99.748	100
Yonsei unlimited	99.91597	100	99.41176	99.83193	100
Yonsei 56k	97.22689	99.4958	96.55462	98.40336	99.91597
Yonsei 10k	99.07563	99.91597	98.23529	99.2437	99.91597
Yonsei 5k	96.38655	99.2437	94.70588	97.31092	99.4958
Yonsei 1k	95.79832	99.2437	93.61345	96.80672	99.32773
Yonsei 0.5k	95.04202	98.90756	91.84874	95.12605	97.73109

Table III.4 – Supplementary error statistics (525 format, 1190 PVSs)

Percentile	Et≤0	Et≤1	Et≤2	Eg≤0.01	Eg≤0.02	Eo≤2.5	Eo≤5.0
NTIA 25k	98.739	99.748	100	82.353	89.916	82.353	94.118
Yonsei unlimited	99.41176	99.91597	100	83.10924	95.71429	91.59664	98.48739
Yonsei 56k	98.23529	99.91597	100	82.26891	94.28571	90.84034	97.89916
Yonsei 10k	98.48739	99.91597	100	81.59664	93.52941	90.2521	97.89916
Yonsei 5k	96.47059	99.57983	99.7479	80.58824	92.43697	90.42017	97.47899
Yonsei 1k	95.88235	99.15966	99.66387	51.42857	75.12605	82.10084	94.78992
Yonsei 0.5 k	93.94958	98.23529	98.7395	47.14286	69.15966	79.41176	93.44538

Table III.5 – Mean values of the five errors (625 format, 1190 PVSs)

	Mean (Ex)	Mean (Ey)	Mean (Et)	Mean (Eg)	Mean (Eo)
NTIA 25k	–0.042016807	–0.058823529	0.304201681	0.003310455	0.270601479
Yonsei unlimited	0.061345	0.018487	0.003361	0.005358	0.350661
Yonsei 56k	0.056303	0.022689	0.012605	0.005665	0.483693
Yonsei 10k	0.052941	0.026891	–0.12101	0.010251	0.901221
Yonsei 5k	0.045378	0.017647	–0.1	0.010353	0.864778
Yonsei 1k	0.086555	0.022689	–0.23109	0.022666	–1.38218
Yonsei 0.5k	0.291597	0.021008	–0.30756	0.036449	–2.527

Table III.6 – Standard Deviation values of the five errors (625 format, 1190 PVSs)

	std (Ex)	std (Ey)	std (Et)	std (Eg)	std (Eo)
NTIA 25k	0.301243791	0.235393043	5.487609638	0.005141177	0.825994079
Yonsei unlimited	0.701148	0.157763	4.785209	0.008276	1.447665
Yonsei 56k	0.705158	0.174938	4.925928	0.008909	1.48402
Yonsei 10k	0.70661	0.194844	4.861287	0.02089	2.677773
Yonsei 5k	0.690893	0.19374	4.648005	0.020985	2.654485
Yonsei 1k	0.900465	0.188811	4.608239	0.022751	3.103119
Yonsei 0.5k	2.500432	0.738509	4.936385	0.037611	4.367109

Table III.7 – Error statistics (625 format, 1190 PVSs)

Percentile	Ex+Ey≤0	Ex+Ey≤1	Ex+Ey+Et=0	Ex+Ey+Et≤1	Ex+Ey+Et≤2
NTIA 25k	92.43697	98.31933	80	84.03361	90.08403
Yonsei unlimited	94.45378	99.07563	84.36975	86.89076	90.33613
Yonsei 56k	94.03361	98.82353	83.94958	86.21849	89.74790
Yonsei 10k	93.52941	98.57143	84.11765	85.96639	90.50420
Yonsei 5k	93.94958	98.48739	84.20168	86.47059	90.08403
Yonsei 1k	93.44538	97.64706	83.78151	86.05042	88.90756
Yonsei 0.5k	90.42017	95.37815	80.58824	83.94958	86.55462

Table III.8 – Supplementary error statistics (625 format, 1190 PVSs)

Percentile	Et≤0	Et≤1	Et≤2	Eg≤0.01	Eg≤0.02	Eo≤2.5	Eo≤5.0
NTIA 25k	80	88.82352	91.428571	88.235294	97.47899	96.63865	100
Yonsei unlimited	85.96639	90.84034	91.26050	86.63866	92.94118	93.02521	96.38655
Yonsei 56k	85.46218	90.58824	90.92437	83.44538	92.18487	92.60504	95.71429
Yonsei 10k	85.71429	91.0084	91.76471	77.05882	86.21849	85.88235	90.42017
Yonsei 5k	86.05042	90.7563	91.59664	77.22689	86.13445	86.38655	91.00840
Yonsei 1k	85.71429	90.84034	91.2605	39.41176	58.9916	64.87395	84.87395
Yonsei 0.5k	83.52941	89.83193	90.7563	32.01681	46.63866	56.55462	76.30252

III.3 Experimental results after removing erroneous video sequences

However, it was found that the original VQEG Phase I data was not correctly calibrated. Consequently, these incorrectly-calibrated video sequences will inevitably introduce additional errors which have nothing to do with the calibration methods. In this analysis, the following scenes, HRCs, and clips were eliminated for the following reasons:

- Discard 625/SRC1 – a still scene.
- Discard 625/HRC14 and 625/HRC1 – the target delay is outside of ±1 second (these 625-line HRCs were miss-generated with delays within ±30f, not 25f).
- Discard HRCs where the gain of truth data is not within ±0.1, or offset of truth data not within ±2.5 (i.e., the original HRC had luminance gain/offset problems).
  - 525/HRC1
  - 525/HRC15
  - 525/HRC16
  - 625/HRC1
  - 625/HRC15
  - 625/HRC16
- Discard individual clips where delay of truth data is not within ±0F.
  - 525 SRC20 HRC13
  - 625 SRC3 HRC15 – (already eliminated above)
  - 625 SRC3 HRC16 – (already eliminated above)
  - 625 SRC2 HRC9

When these erroneous video sequences are removed, there are 973 PVSs for the 525 video format and 812 PVS for the 625 video format. Tables III.9-11 show the mean & standard deviation values of the five errors and error statistics for the 525 video format after removing the erroneous video sequences.

Table III.9 – Mean values of the five errors (525 format, 973 PVSs)

	Mean (Ex)	Mean (Ey)	Mean (Et)	Mean (Eg)	Mean (Eo)
NTIA 25k	0	0	0	0.001435005	0.342328695
Yonsei unlimited	0	0	0	0.002762	0.245746
Yonsei 56k	0	0	0	0.002821	0.281161
Yonsei 10k	0	0	0.002055	0.002955	0.281523
Yonsei 5k	0	0	0	0.003149	0.275845
Yonsei 1k	0	0	–0.00514	0.011791	–0.55751
Yonsei 0.5k	–0.00411	–0.00206	–0.02569	0.014032	–0.65768

Table III.10 – Standard deviation values of the five errors (525 format, 973 PVSs)

	std (Ex)	std (Ey)	std (Et)	std (Eg)	std (Eo)
NTIA 25k	0	0	0	0.00174774	0.371539333
Yonsei unlimited	0	0	0	0.002658	0.489556
Yonsei 56k	0	0	0	0.003458	0.500722
Yonsei 10k	0	0	0.045314	0.003683	0.510453
Yonsei 5k	0	0	0.090722	0.004088	0.533234
Yonsei 1k	0	0	0.146896	0.011847	1.555617
Yonsei 0.5k	0.090629	0.045314	0.430759	0.01483	1.800581

Table III.11 – Error statistics (525 format, 973 PVSs)

Percentile	Ex+Ey≤0	Ex+Ey≤1	Ex+Ey+Et=0	Ex+Ey+Et≤1	Ex+Ey+Et≤2
NTIA 25k	100	100	100	100	100
Yonsei unlimited	100	100	100	100	100
Yonsei 56k	100	100	100	100	100
Yonsei 10k	100	100	99.79445	100	100
Yonsei 5k	100	100	99.48613	99.89723	100
Yonsei 1k	100	100	99.07503	99.5889	100
Yonsei 0.5k	99.79445	99.79445	97.73895	99.1778	99.48613

Table III.12 – Supplementary error statistics (525 format, 973 PVSs)

Percentile	Et≤0	Et≤1	Et≤2	Eg≤0.01	Eg≤0.02	Eo≤2.5	Eo≤5.0
NTIA 25 k	100	100	100	100	100	100	100
Yonsei unlimited	100	100	100	98.04728	100	100	100
Yonsei 56k	100	100	100	96.3001	99.5889	100	100
Yonsei 10k	99.79445	100	100	95.68345	99.38335	100	100
Yonsei 5k	99.48613	99.89723	100	93.93628	98.7667	100	100
Yonsei 1k	99.07503	99.5889	100	56.21788	80.57554	87.46146	98.3556
Yonsei 0.5k	97.9445	99.38335	99.69168	52.82631	74.61459	85.30319	97.1223

Tables III.13-16 show the mean values of the five errors and error statistics for the 625 video format after removing the erroneous video sequences.

Table III.13 – Mean values of the five errors (625 format, 812 PVSs)

	Mean (Ex)	Mean (Ey)	Mean (Et)	Mean (Eg)	Mean (Eo)
NTIA 25k	0	0	–0.004926108	0.001013967	0.157564719
Yonsei unlimited	0	0	0	0.002403	0.154763
Yonsei 56k	0	0	0	0.002885	0.224836
Yonsei 10k	0	0	0	0.00305	0.223728
Yonsei 5k	0	0	0	0.003139	0.231451
Yonsei 1k	0	0	0	0.01779	–1.61664
Yonsei 0.5k	0.107143	0	–0.00862	0.030812	–2.71628

Table III.14 – Standard deviation values of the five errors (625 format, 812 PVSs)

	std (Ex)	std (Ey)	std (Et)	std (Eg)	std (Eo)
NTIA 25k	0	0	0.070056306	0.0012045	0.292369567
Yonsei unlimited	0	0	0	0.002755	0.464458
Yonsei 56k	0	0	0	0.003775	0.563434
Yonsei 10k	0	0	0	0.003861	0.570579
Yonsei 5k	0	0	0	0.003983	0.585887
Yonsei 1k	0	0	0	0.018632	2.136707
Yonsei 0.5k	1.841325	0	0.160685	0.033967	3.775601

Table III.15 – Error statistics (625 format, 812 PVSs)

Percentile	Ex+Ey≤0	Ex+Ey≤1	Ex+Ey+Et=0	Ex+Ey+Et≤1	Ex+Ey+Et≤2
NTIA 25k	100	100	99.50738916	100	100
Yonsei unlimited	100	100	100	100	100
Yonsei 56k	100	100	100	100	100
Yonsei 10k	100	100	100	100	100
Yonsei 5k	100	100	100	100	100
Yonsei 1k	100	100	100	100	100
Yonsei 0.5k	97.78325	98.64532	97.53695	98.64532	98.64532

Table III.16 – Supplementary error statistics (625 format, 812 PVSs)

Percentile	Et≤0	Et≤1	Et≤2	Eg≤0.01	Eg≤0.02	Eo≤2.5	Eo≤5.0
NTIA 25k	99.507389	100	100	100	100	100	100
Yonsei unlimited	100	100	100	97.04433	99.50739	100	100
Yonsei 56k	100	100	100	92.61084	99.38424	99.50739	100
Yonsei 10k	100	100	100	92.85714	98.76847	99.50739	100
Yonsei 5k	100	100	100	93.2266	98.89163	99.26108	100
Yonsei 1k	100	100	100	46.67488	66.99507	73.02956	91.62562
Yonsei 0.5k	98.52217	99.63054	100	35.59113	52.0936	61.2069	79.18719

III.4 Experimental results per HRC after removing erroneous video sequences

HRC analysis was performed by median filtering over all scenes (SRC) for each HRC/case. For each HRC & case (1-7), we pool all luminance gain values and use the median gain value as the HRC gain value. For each HRC & case (1-7), we pool all horizontal (x) shift values and use the median horizontal shift value as the HRC horizontal shift value. Likewise, we apply this median filter to luminance offset, vertical shift, and delay. These calculations were performed after discarding the erroneous video sequences identified above. This results in 82 HRCs for 525-line and 91 HRCs for 625-line.

Table III.17 – Mean values of the five errors after removing the erroneous video sequences (525 format, 82 HRCs)

	Mean (Ex)	Mean (Ey)	Mean (Et)	Mean (Eg)	Mean (Eo)
NTIA 25k	0	0	0	0.001422593	0.325705268
Yonsei unlimited	0	0	0	0.002762	0.245746
Yonsei 56k	0	0	0	0.002191	0.262284
Yonsei 10k	0	0	0	0.002241	0.264225
Yonsei 5k	0	0	0	0.002211	0.269888
Yonsei 1k	0	0	0	0.009496	–0.49409
Yonsei 0.5k	0	0	0	0.010771	–0.49853

Table III.18 – Standard deviation values of the five errors after removing the erroneous video sequences (525 format, 82 HRCs)

	std (Ex)	std (Ey)	std (Et)	std (Eg)	std (Eo)
NTIA 25k	0	0	0	0.001802808	0.386266756
Yonsei unlimited	0	0	0	0.002658	0.489556
Yonsei 56k	0	0	0	0.001162	0.340323
Yonsei 10k	0	0	0	0.001202	0.335712
Yonsei 5k	0	0	0	0.001243	0.330446
Yonsei 1k	0	0	0	0.006518	0.780895
Yonsei 0.5k	0	0	0	0.007212	0.745808

Table III.19 – Error statistics after removing the erroneous video sequences (525 format, 82 HRCs)

Percentile	Ex+Ey≤0	Ex+Ey≤1	Ex+Ey+Et=0	Ex+Ey+Et≤1	Ex+Ey+Et≤2
NTIA 25k	100	100	100	100	100
Yonsei unlimited	100	100	100	100	100
Yonsei 56k	100	100	100	100	100
Yonsei 10k	100	100	100	100	100
Yonsei 5k	100	100	100	100	100
Yonsei 1k	100	100	100	100	100
Yonsei 0.5k	100	100	100	100	100

Table III.20 – Supplementary error statistics after removing the erroneous video sequences (525 format, 82 HRCs)

Percentile	Et≤0	Et≤1	Et≤2	Eg≤0.01	Eg≤0.02	Eo≤2.5	Eo≤5.0
NTIA 25k	100	100	100	100	100	100	100
Yonsei unlimited	100	100	100	100	100	100	100
Yonsei 56k	100	100	100	100	100	100	100
Yonsei 10k	100	100	100	100	100	100	100
Yonsei 5k	100	100	100	100	100	100	100
Yonsei 1k	100	100	100	57.31707	91.46341	98.78049	100
Yonsei 0.5k	100	100	100	56.09756	89.02439	100	100

Table III.21 – Mean values of the five errors after removing the erroneous video sequences (625 format, 91 HRCs)

	Mean (Ex)	Mean (Ey)	Mean (Et)	Mean (Eg)	Mean (Eo)
NTIA 25k	0	0	0	0.0010142	0.156838637
Yonsei unlimited	0	0	0	0.001885441	0.145601275
Yonsei 56k	0	0	0	0.00213	0.175956
Yonsei 10k	0	0	0	0.002283	0.174766
Yonsei 5k	0	0	0	0.002412	0.18998
Yonsei 1k	0	0	0	0.015044	–1.30766
Yonsei 0.5k	0	0	0	0.025443	–1.92406

Table III.22 – Standard deviation values of the five errors after removing the erroneous video sequences (625 format, 91 HRCs)

	std (Ex)	std (Ey)	std (Et)	std (Eg)	std (Eo)
NTIA 25k	0	0	0	0.001205754	0.293524551
Yonsei unlimited	0	0	0	0.00157242	0.248337763
Yonsei 56k	0	0	0	0.00184	0.241735
Yonsei 10k	0	0	0	0.001812	0.239511
Yonsei 5k	0	0	0	0.001978	0.265361
Yonsei 1k	0	0	0	0.011339	0.982579
Yonsei 0.5k	0	0	0	0.020609	1.802244

Table III.23 – Error statistics after removing the erroneous video sequences (625 format, 91 HRCs)

Percentile	Ex+Ey≤0	Ex+Ey≤1	Ex+Ey+Et=0	Ex+Ey+Et≤1	Ex+Ey+Et≤2
NTIA 25k	100	100	100	100	100
Yonsei unlimited	100	100	100	100	100
Yonsei 56k	100	100	100	100	100
Yonsei 10k	100	100	100	100	100
Yonsei 5k	100	100	100	100	100
Yonsei 1k	100	100	100	100	100
Yonsei 0.5k	100	100	100	100	100

Table III.24 – Supplementary error statistics after removing the erroneous video sequences (625 format, 91 HRCs)

Percentile	Et≤0	Et≤1	Et≤2	Eg≤0.01	Eg≤0.02	Eo≤2.5	Eo≤5.0
NTIA 25k	100	100	100	100	100	100	100
Yonsei unlimited	100	100	100	100	100	100	100
Yonsei 56k	100	100	100	100	100	100	100
Yonsei 10k	100	100	100	98.9011	100	100	100
Yonsei 5k	100	100	100	97.8022	100	100	100
Yonsei 1k	100	100	100	35.16484	79.12088	87.91209	100
Yonsei 0.5k	100	100	100	19.78022	42.85714	71.42857	96.7033

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	**Cable networks and transmission of television, sound programme and other multimedia signals**
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2009