ISO/IEC JTC 1/SC 29

Coding of audio, picture, multimedia and hypermedia information

Secretariat: JISC (Japan)

Document type:	Text for PDTR ballot or comment
Title:	Text of ISO/IEC PDTR 29170-1.2: Information technology -- Advanced image coding and evaluation methodologies -- Part 1: Guidelines for codec evaluation [SC 29/WG 1 N 72032]
Status:	Text of PDTR agreed at the 72nd SC29/WG 1 Meeting (Ref. SC 29 N 15906, 37). In accordance with Resolution 1 taken at the 29th SC 29 Plenary Meeting, 2016-06-04, Geneva, Switzerland, the SC 29 Secretariat issued 2nd PDTR ballot. [Requested action: For ballot by SC 29 P-members]
Date of document:	2016-07-22
Source:	ISO/IEC JTC 1/SC 29/WG 1
Expected action:	VOTE
Action due date:	2016-09-21
No. of pages:	37
Email of secretary:	sc29-sec@itscj.ipsj.or.jp
Committee URL:	http://isotc.iso.org/livelink/livelink/open/jtc1sc29

ISO/IEC JTC 1/SC 29/Nxxxxx

Date: 2016-06-3

ISO/IEC PDTR 29170-1

ISO/IEC JTC 1/SC 29/WG 1

Secretariat: JISC

ISO/IEC JTC1 / SC 29 / WG 1
INTERNAL DOCUMENT

ONLY FOR MEMBERS AND AUTHORIZED DISTRIBUTION BY LIAISONS

Information technology — Advanced image coding and evaluation — Part 1: Guidelines for image coding system evaluation

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: Technical Report
Document subtype:
Document stage: (40) Enquiry
Document language: E

D:\temp\macroserver\DOCX2PDFISOTC\DOCX2PDFISOTC.lliadmin@SRVWEB100_1521\17801435_1.doc
STD Version 2.1c2

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, the joint technical committee may propose the publication of a Technical Report of one of the following types:

— type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;

— type 3, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard (“state of the art”, for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC TR 29170-1, which is a Technical Report of type 3, was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Advanced image coding and evaluation*.

ISO/IEC TR 29170 consists of the following parts, under the general title *Information technology — Advanced image coding and evaluation*:

— *Part 1: Guidelines for image coding system evaluation*

— *Part 2: Evaluation procedure for nearly lossless coding*

Introduction

This technical report provides a framework and best practices to evaluate image compression algorithms. This document provides a selection of evaluation tools that allow testing multiple features including objective metric image quality, subjective metric image and codec algorithmic complexity. Which features of codecs under evaluation that should be tested and pass-fail criteria is beyond the scope of this document.

Information technology — Advanced image coding and evaluation — Part 1: Guidelines for image coding system evaluation

1 Scope

This technical report recommends best practices for coding system evaluation of images and image sequences. This report defines a common vocabulary of terms for coding system evaluation and divides evaluation methods into three broad categories below.

1) Subjective assessment

2) Objective assessment

3) Computational assessment

In addition to these broad assessment categories, this guideline discussed special care that is given for coding unusual imagery, for example, high dynamic range or high colour depth.

A fourth assessment category, hardware complexity, is often important for real-time or computationally complex applications, however, it outside the scope of this technical report.

2 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

2.1
compressor
portion of a coding system (codec) that has a pixel stream and may have control metadata as its input and a coded bitsream as its output

2.2
component bit depth
the number of bits of precision of colour channels (or components) of the unenclosed image

2.3
component number
the number of colour channels (or components) encoded in an image

2.4
constant bit rate
mode where the number of encoded bits from a portion of an image represented by a fixed number of pixels does not vary compared to the number of encoded buts in any other equally sized portion of the same image

2.5
decompressor
portion of a coding system (codec) that has a coded bitstream as its input and a pixel stream as its output

2.6
drift
the net generational loss of image quality if the output of a lossy image compression/reconstruction cycle is recompressed again under the same conditions by the same codec

2.7
expert observer
an observer that has expertise in the image artefacts that may be introduced by the system under test or who has designed or participated in the selection of test content for the system under test.

2.8
horizontal pixel resolution
horizontal extent of the image in image pixels where the horizontal extent may depend on the channel.

2.9
idempotent
a codec that operates lossless on its own decompression output

2.10
non-expert observer (naïve observer)
an observer that has no expertise in the image artefacts that may be introduced by the system under test

2.11
objective assessment
a computational algorithmic process leading to a numerical score for all or a portion of an image under test

2.12
quality loss
{provide a definition from the recursive test section}

2.13
sample precision
the sample precision describes the bit depth of a given data type encoding the image.

2.14
sample type
the type of numeric value that contains sample values to a resolution specified by sample precision where types can include unsigned integers, signed integers and floating point or fixed point samples

2.15
subjective assessment
an algorithmic process where recorded observations from human subjects (observers) lead to a numerical score for all or a portion of an image under test

2.16
variable bit rate
mode where the number of encoded bits in a portion of a image represented by a fixed number of pixels can be different from the number of encoded bits in any other equally sized portion of the same image

2.17
vertical pixel resolution
vertical extent of the image in pixels and the vertical extent may depend on the channel for subsampled images

3 Symbols (and abbreviated terms)

BPP: bits per pixel

CR: compression ratio

CSF: contrast sensitivity function

HDR: high dynamic range

LDR: low dynamic range, synonymous with SDR

MOS: mean opinion score

MSE: mean squared error

PSNR: peak signal-to-noise ratio

SDR: standard dynamic range, synonymous with LDR

SSIM: structural similarity index

VDP: visual differences predictor

4 Selection and characteristics of test images

4.1 Common image characteristics

Image selection relies on a common vocabulary for describing image characteristics. This section defines this vocabulary and the applicability to testing both standard and high dynamic range images.

For example, integer samples in range [0..1023] are here described as ten bit data, regardless of whether the samples are stored in 16 bit values or packed into ten bits each. Integer values in the range [-128..127] are here classified as 8 bit signed data because the data representation consists of one sign bit and seven magnitude bits.

The image dimension data consists of the full set of data defined above, that is, the number of channels, the width and height of each image channel, the sample type of each channel and the sample precision of each channel.

4.2 Bits per pixel

Bits Per Pixel (BPP) describes the compression performance of image compression codecs independent of the original image's sample size.

BPP is defined independently of the image sample precision as the size of the compressed image stream *L* in bytes and the Image Dimensions (see B.1):

$$BPP = \frac{8 \cdot L}{w \cdot h} \qquad \textbf{Eq. 1}$$

4.3 Compression ratio

Compression ratio (CR) describes the compression performance of image coding system dependent of the original image's sample size.

$$CR = \frac{\sum_{c=0}^{d-1} b(c) \cdot w(c) \cdot h(c)}{8 \cdot L} \qquad \text{Eq. 2}$$

The quantity *d* is the number of channels of the image, *w(c)* is the horizontal extend of channel *c* and *h(c)* the vertical extent of the channel. The number *b(c)* is the number of bits required to describe the samples of channel *c*.

4.4 Variation in bit rates

4.4.1 Constant bit rate systems

By the definition of the coding system, variations do not occur in the coded bit rate within an image, A test can verify if any bit rate variation is present. This restriction does not apply between two or more images or images.

4.4.2 Variable bit rate systems

For some applications, it is important that a coding system is able to generate a continuous stream of symbols, ensuring that some output is generated at least in every given time span, i.e. that the output bitrate does not vary too much over time. For example, carry-over resolution in arithmetic coding might cause arbitrary long delays in the output until the carry can be resolved.

For the purpose of this test, the output bitrate is defined as the number of output symbols generated for each input symbol, measured in dependence of the percentage of the input stream fed into the codec. If a compressor can generate parts of its output from incomplete input, it is said to be *online-capable*. Only for such codecs this test is defined.

A measurement procedure to measure bit rate variations appears in Annex D.

4.5 Error resilience

The ability for a system to withstand or recover from errors is orthogonal to the coding system quality. In modern systems, error resiliency can be assisted by error markers in the bitstream, but error resiliency can also be part of transport layer assessments. A coding system evaluation needs to take into consideration whether error resiliency is in a bitstream and if so whether optional or intertwined and inseparable.

The best practices at the time of this recommendation separates error resiliency by computing the efficiency of the algorithm to code images while assuming a perfect transmission medium. The ability to recover errors can be added either through resiliency markers, forward error correction or merely parity checking to identify but not correct errors. As such, this topic is outside the scope of this recommendation.

If error markers and error handling is not separable, the coding system efficiency will include such markers.

4.6 Recursive compression assessment

4.6.1 Definition

Generation loss is a loss in image quality if the output of a lossy image compression/reconstruction cycle is recompressed again under the same conditions by the same. If this recompression is repeated over several cycles, severe degradation of image quality can result.

Generation loss limits the number of repeated compressions/decompressions in an image processing chain if repeated recompression generates severely more distortion than a single compression/decompression cycle. This sub-clause distinguishes between drift and quality loss. While the former is due to a systematic DC error often due to mis-calibration in the colour transformation or quantisation, the latter covers all other error sources as well, as for example due to limited precision in the image transformation implementation.

A measurement procedure to measure generational quality loss appears in Annex D.

4.7 Image selection

The coding system should in general operate over many image categories or image types. Some of these are continuous tone images, some may simply be black and white or half tones. Test material should reflect the potential applications in which a coding system will be used. The following examples are common bases for image evaluation categories.

1) Natural scenes

2) Portraits and candids, several with differing skin tones

3) Compound (multi-layer)

4) Photo-realistic synthetic

5) Graphics and animations

6) Text and web pages

7) Engineered test patterns

If the coding system is intended for specific image types or applications, such as medical imaging, a set of images appropriate to the application should be the test set.

Image size used during testing should be appropriate for the application, not very much smaller or larger than targeted in typical usage.

5 Best practices of subjective image quality assessments

5.1 Goals of subjective assessment

Some subjective image assessment methods are likely to reflect the human notion of quality by anticipating *the reactions of those who might view the tested systems.* While other subjective image assessment methods can determine if some artefacts are visually discernible and likely to adversely affect image quality. These methods become the best quality assessment methods. However, they are very time demanding and eventually they might become very expensive, because of the cost of the viewers and also of the system under test implementation.

Test evaluations can be application specific, for example, according to BT.500,

> *"Subjective assessment methods are used to establish the performance of television systems using measurements that more directly anticipate the reactions of those who might view the systems tested. In this regard, it is understood that it may not be possible to fully characterize system performance by objective means; consequently it is necessary to supplement objective measurements with subjective measurements."*

This report suggests that best practice should separate applications from the image quality evaluation to the best extent possible. Subjective assessment methodology recommended herein follows this guideline.

5.2 Subjective assessment evaluation procedures

5.2.1 Observer selection

Evaluators should prefer naïve observers for most general viewing or entertainment applications. In the case of specialized imaging, such as, medical or structural engineering, an expert observer who can discern defects from artefacts is needed.

5.2.2 Visual acuity

Common to all subjective evaluation procedures, observers will need to demonstrate meet a well-defined visual acuity. Sometimes colour vision is not tested.

The following recommendations usually apply.

1) Test for visual acuity with or without corrective lens, either glasses or contacts that do not have multiple focal lengths, e.g., progressive, bifocal or trifocal corrective lens.

2) Verify normal visual acuity by using a Snellen or Landolt test charts where the observer reads at 20/20 from 50 cm.

3) If screening for normal colour vision, verify by testing with Ishihara plates or equivalent.

Evaluators may refer to ISO/IEC 29170-2 for example of tools that help assess an observer's visual acuity.

5.2.3 Number of observers

The number of observers is dependent on the evaluation system. For example, according to BT.500,

> *"At least 15 observers should be used. The number of assessors needed depends upon the sensitivity and reliability of the test procedure adopted and upon the anticipated size of the effect sought. For studies with limited scope, e.g., of exploratory nature, fewer than 15 observers may be used. In this case, the study should be identified as 'informal'. The level of expertise in television image quality assessment of the observers should be reported."*

The example from ISO/IEC 29170-2, casts more importance on repetitions per observer and less on observer number. These guidelines for the observer population apply:

> *"The observer population should include variations in gender, ethnicity and age. The experiment is visual in nature and age can strongly correlate to visual acuity, therefore, participant age for this procedure favours the age range for the observer from 18 to 30 years old.*

In some cases, a evaluation procedure may set an absolute age limit due to visual acuity degradation with age. For example, ISO/IEC 29170-2 limits an observer's age to 40 years or less."

5.2.4 Instructions to observers

Each procedure should contain directions for observer instruction. In general, the procedure should be understood, when to take breaks, how to use any applicable user interface or software tools. In the event of grading, explain the relative scale and illustrate with examples of good and impaired images of various types.

5.2.5 Evaluation scales

Subjective testing usually employs one of the following scales: Lickert scale, (Rec. ITU-R BT.500 and ITU-T P.910), Quality ruler (ISO 20462-3) and forced choice and ternary choice procedures (ISO/IEC 29170-2 and BT.500).

Refer to BT.500 for an explanation of assessment problems and methods used in television. P.910 was used successfully for teleconferencing systems quality analysis.

P.910 also cites usage of an explicit reference, depending on the objective of the testing.

> *"An important issue in choosing a test method is the fundamental difference between methods that use explicit references (e.g., DCR), and methods that do not use any explicit reference (e.g., ACR, ACR-HR, and PC). This second class of method does not test transparency or fidelity.*

5.2.6 Statistical analysis

This section recommends several methods for statistical analysis, each represent a separate topic. For information about mean opinion score calculation and data treatment, refer to Annex A.

> *"Because they vary with range, it is inappropriate to interpret judgements from most of the assessment methods in absolute terms (e.g. the quality of an image or image sequence).*
>
> *"For each test parameter, the mean and 95% confidence interval of the statistical distribution of the assessment grades must be given. If the assessment was of the change in impairment with a changing parameter value, curve-fitting techniques should be used. Logistic curve-fitting and logarithmic axis will allow a straight line representation, which is the preferred form of presentation."* (BT.500)

This report also refers readers to ISO/IEC 29170-2 Annex D for statistical treatment of binary and ternary forced choice data reports.

5.3 Viewing conditions for electronic displays

5.3.1 Purpose

Various international standards and guidelines from trade organizations exist that are relevant to compression investigators. This section describes some of the viewing conditions arranged for viewing in standards defined by ISO and ISO/IEC and other references related closely with the end application, such as, home television viewing or an office work environment.

5.3.2 ISO 3664

Originally designed for photographs, the ISO 3664:2009 international standard defines viewing conditions for laboratory testing environments. This is useful for native compression evaluation without distractions and other influences from surrounding light.

However, any viewing conditions procedure is debatable in a fixed environment. The ideal conditions for evaluation of an entire display system may be in the environment where it will be used or the photograph viewed (cit. ISO 3664). White points will vary, recommendations include:

1) Colour electronic displays: D65

2) Television: D65

3) Photographs: D50

For evaluation of compression systems on colour monitors, this guideline recommends adherence to the methods in ISO 3664 in all practical aspects. Deviations either can be defined in an applicable standard or noted in a test report. In all cases for subjective evaluations, test reports should take care to report sufficient detail for an evaluator skilled in the art to recreate applicable testing and viewing conditions.

5.3.3 ISO 9241

The ISO 9241 family of standards defines viewing conditions and ergonomic conditions of office viewing monitor, takes into consideration many factors including ambient lighting, viewing distance, viewer's age and so forth. The standard represents a large body of work that can serve as a useful reference for the compression expert when evaluating or designing suitable coding systems for the office environment.

5.4 Goals for evaluation of visually lossless and nearly lossless coding

The ISO/IEC 29170-2 *Evaluation procedure for nearly visually lossless coding* is useful for evaluating lightly compressed coding systems. For instance, display stream compression where a source compresses image data sent to a display may be evaluated as visible or invisible to a viewer. Examples of display streams include but are not limited a wired link between a set-top box unit and a television or between a mobile host graphics processor to a display panel module in a mobile appliance.

A coding system will be considered visually lossless if the test results obtained when using this procedure meet a pre-defined acceptable quality level. The interpretation of data obtained by this subjective test procedure that may lead to a pass-fail threshold is outside the scope of this report.

The procedure compares individual images with various binary or ternary forced choice protocols. The procedure also relies only on subjective evaluation methods designed to discern image imperfections on electronic colour displays of any technology or size.

6 Best practices of objective image quality assessment methodology

In the recent literature many papers have proposed objective quality metrics dedicated to several image and video applications. This report recommends a few well-known metrics and a set of best practices.

Objective evaluation metrics can be categorized into three groups:

1) full-reference

2) no-reference

3) reduced-reference

Full reference metrics need full information of the original images and demand ideal images as references which can be hardly achieved in practice. The traditional methods (such as peak signal-to-noise-ratio PSNR) are based on pixel-wise error and have not always been in agreement with perceived quality measurement. Recently, some full reference metrics modelled by simulating the human visual system have been proposed. For instance, Wang et al. introduced in an alternative complementary framework for quality assessment based on the degradation of structural information. They developed a structural similarity index (SSIM) and demonstrated its promise through a set of intuitive examples.

No reference metrics aim to evaluate distorted images without any cue from their original ones. "No reference" coding evaluation tends not to be favoured by this report because most of the proposed no reference quality metrics are designed for one or sets of predefined specific distortion types and are unlikely to be generalized for evaluating images degraded with other types of distortions.

Reduced reference metrics make use of a part of the information from the original images in order to evaluate the visual perception quality of the distorted ones. As the transformed and stored data are reduced, reduced reference metrics have a great potential in some specific applications of quality assessment.

Best practice recommendations appear in Annex B, which contains entirely full reference algorithms. The Annex contains objective metrics that this technical working group has found useful when comparing codecs designed within ISO/IEC and those from other organizations. As such, this collection represents an understanding of best and common practice.

Annex A

Subjective metrics

A.1 Mean opinion score

The Mean Opinion Score (MOS) provides a numerical indication of the perceived quality of an image or an image sequence after a process such as compression, quantization, and transmission. The MOS is expressed as a single number in the range 1 to 5 in the case of a discrete scale (resp. 1 to 100 in the case of a continuous scale), where 1 is the lowest perceived quality, and 5 (resp. 100) is the highest perceived quality. Its computation allows to study the general behaviour of the observers with regards to a given impairment.

A.1.1 MOS calculation

The interpretation of the obtained judgments is completely dependent on the nature of the constructed test. The MOS $\overline{m}_{jkr}$ is computed for each presentation:

$$\overline{m}_{jkr} = \frac{1}{N}\sum_{i=1}^{N} m_{ijkr} \qquad \textbf{Eq. 3}$$

where $\overline{m}_{ijkr}$ is the score of the observer *i* for the degradation *j* of the image *k* and the r^{th} iteration. *N* represents the number of observers. In a similar way, we can calculate the global average scores, $\overline{m}_{j}$ and $\overline{m}_{k}$, respectively for each test condition (impairment) and each test image.

A.1.2 Calculation of confidence interval

In order to evaluate as well as possible the reliability of the results, a confidence interval is associated to the MOS. It is commonly adopted that the 95% confidence interval is enough. This interval is designed as:

$$\left[\overline{m}_{jkr} - \delta_{jkr}, \overline{m}_{jkr} + \delta_{jkr}\right] \qquad \textbf{Eq. 4}$$

where :

$$\delta_{jkr} = 1.95 \frac{s_{jkr}}{\sqrt{N}} \qquad \textbf{Eq. 5}$$

s_{jkr} represents the standard deviation defined as:

$$s_{jkr} = \sqrt{\sum_{i=1}^{N} \frac{(\overline{m}_{jkr} - m_{ijkr})^2}{N-1}} \qquad \textbf{Eq. 6}$$

A.1.3 Outliers rejection

One of the objectives of results analysis is also to be able to eliminate from the final calculation either a particular score, or an observer. This rejection allows to correct influences induced by the observer's behaviour, or bad choice of test images. The most obstructing effect is incoherence of the answers provided by an observer, which characterizes the non-reproducibility of a measurement. The ITU-R 500-10 recommendation contains a way to reject incoherent results.

To that aim, it is necessary to calculate the MOS and the standard deviations associated with each presentation. These average values are functions of two variables the presentations and the observers. Then, check if this distribution is normal by using the β_2 test. The latter is the kurtosis coefficient (i.e. the ratio between the fourth-order moment and the square of the second-order moment). Therefore, the $\beta_{2\ jkr}$ to be tested is given by:

$$\beta_{2\ jkr} = \frac{\frac{1}{N}\sum (\overline{m}_{jkr} - m_{ijkr})^4}{\left(\frac{1}{N}\sum (\overline{m}_{jkr} - m_{ijkr})^2\right)^2} \qquad \textbf{Eq. 7}$$

If $\beta_{2\ jkr}$ is between 2 and 4, we can consider that the distribution is normal. In order, to compute P_i and Q_i values allowing taking the final decision regarding the outliers, the observations $\overline{m}_{ijkr}$ for each observer i, each degradation j, each image k, and each iteration r, is compared thanks to a combination of the MOS and the associated standard deviation. The different steps of the algorithm are summarized in the following algorithm.

Algorithm 1 : Steps for outliers rejection

```
if (2 ≤ β2jkr ≤ 4) /* (normal distribution) */ then
    if (uijkr ≥ ūijkr + 2σjkr) then
        Pi = Pi + 1;
    endif
    if (uijkr ≤ ūijkr − 2σjkr) then
        Qi = Qi + 1;
    endif
endif
else
    if (uijkr ≥ ūijkr + √20 σjkr) then
        Pi = Pi + 1;
    endif
    if (uijkr ≤ ūijkr − √20 σjkr) then
        Qi = Qi + 1;
    endif
endif
/* Finally, we can carry out the following rejection test : */
if ((Pi+Qi)/(J.K.R) > 0.05) and (|(Pi−Qi)/(Pi+Qi)| < 0.3) then
    Eliminate scores of observer i;
endif
/* Where J is the total number of degradations, K is the total number of images and R is the total number
of iterations. */
```

A.2 Binary forced choice image comparison for nearly lossless imagery

Refer to Annex D of ISO/IEC 29170-2 for statistical treatment of binary and ternary forced choice experimental design.

Annex B

Objective metrics

B.1 Mean squared error

Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) approximate image quality in a full reference quality assessment framework. However, PSNR and MSE correlate poorly to to perceived quality, therefore, should be used with care when comparing coding systems or a change of parameters for one coding system.

Record the mean square error between the original and the reconstructed image. Denote the sample value of the reference image at position *x,y* in channel *c* by *p(x,y,c)* and the sample value of the reconstructed image in channel *c* at position *x,y* by *q(x,y,c)*. Denote by *d* the number of image channels, the width of channel *c* by *w(c)* and its height by *h(c)*. Then the Mean Square Error between the reference and the reconstructed image is defined as follows:

$$MSE = \frac{1}{d}\sum_{c=0}^{d-1}\frac{1}{w(c)\cdot h(c)}\sum_{x=0}^{w(c)-1}\sum_{y=0}^{h(c)-1}(p(x,\ y,\ c)-q(x,\ y,\ c))^2$$

B.2 Peak signal to noise ratio

PSNR (Peak Signal to Noise Ratio) is a quantity related to the Mean Square Error and defined as follows: Let *c* denote the image channel, *t(c)* the sample type of this channel and *b(c)* the sample precision of this channel (see B.1). Then define the quantity *m(c)* as follows:

t(c) = signed or unsigned integers	m(c) = $2^{b(c)}$-1
t(c) = floating point or fixed point	m(c) = 1

The PSNR is then:

$$PSNR = -10\log_{10}\left(\frac{1}{d}\sum_{c=0}^{d-1}\frac{\sum_{x=0}^{w(c)-1}\sum_{y=0}^{h(c)-1}(p(x,\ y,\ c)-q(x,\ y,\ c))^2}{w(c)\cdot h(c)\cdot m(c)^2}\right) \qquad \text{Eq. 8}$$

NOTE: The purpose of this measurement is not to define an image quality. A separated benchmark exists for this test. It is rather designed to identify pathological cases where incorrect or unreasonable compressed streams are generated.

B.3 Structural similarity index

B.3.1 SSIM

The Structural Similarity Index (SSIM) proposed by Wang et al. [2004] quantifies the visible difference between a distorted image and a reference image. This index is based on the UIQ [Wang and Bovik, 2002]. The algorithm identifies the structural information in an image as those attributes that represent the structure of the objects in the scene, independent of the average luminance and contrast. The index is based on a combination of luminance, contrast and structure comparison. The comparisons are done for local windows in the image; the overall image quality is the mean of all these local windows.

$$SSIM(X,Y) = \frac{1}{N}\sum_{i=1}^{N} ssim(x_i, y_i)$$

where X and Y are the reference and distorted images, x_i and y_i are images content in a local window *i* and *N* indicates the total number of local windows. Figure X shows the SSIM flowchart, where signal x or signal y has perfect quality and the other is the distorted image.

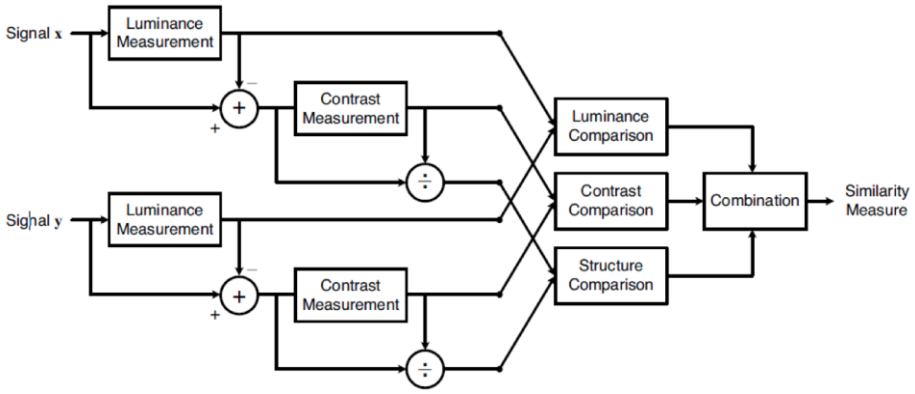

Figure B.1 — Flowchart of the SSIM metric

Several values, for example, block size and block overlap, potential colour weights, applied in tests by the original SSIM metric were left undefined which may make cross correlation of results undependable. Use and documentation of SSIM should clarify undefined terms and parameters.

B.3.2 Multiscale SSIM

A multiscale version of SSIM was proposed by Wang et al. The original and reproduction is run through the SSIM, where the contrast and structure is computed for each subsampled level. The images are low-passed filtered and down-sampled by 2. The lightness (l) is only computed in the final step, contrast (c) and structure (s) for each step. The overall values are obtained by multiplying the lightness value with the sum of contrast and structure for all subsampled levels. Weighting parameters for l, c and s are suggested based on experimental results.

B.3.3 Complex wavelet SSIM

Wang and Simoncelli address the problem SSIM has with translation, scaling and rotation. The solution for this is to extend SSIM to the complex wavelet domain. In order to apply Complex Wavelet SSIM (CWSSIM) for comparing images, the images are decomposed using a complex version of the steerable pyramid transform.

The CWSSIM is computed with a sliding window, and the overall similarity is estimated as the average of all local CWSSIM values. From the objective test done, CWSSIM outperform SSIM and MSE. The authors also tested the metric as a similarity measure on 2430 images.

B.4 Visual difference predictors

B.4.1 Overview

The VDP, VDP2 and HDR-VDP2 are full reference image quality metrics using a model of human visual system (HVS) to quantify the difference between original and test image. Its main advantage, compared to other FR measures, is the ability to compare images with both standard dynamic range (SDR) and high dynamic range (HDR).

The model includes simulation of optical retinal pathway with intra-ocular light scatter, photoreceptor spectral sensitivity, luminance masking, and achromatic response followed by multi-scale decomposition and neural noise model containing neural contrast sensitivity function (CSF) and contrast masking. It requires setting of several parameters about display and viewing conditions.

The result is a visibility map showing the probabilities of difference detection. This could be pooled to provide a single difference visibility value or quality mean opinion score (MOS). The newest version of the metric – HDR-VDP-2.2 **Error! Reference source not found.** – is calibrated on the representative set of SDR and HDR images to provide more reliable MOS estimations.

B.4.2 VDP

This is an algorithm proposed by Daly. The goal of the Visible Differences Predictor (VDP) is to determine the degree to which physical differences become visible differences. The author states that this is not an image quality metric, but it addresses the problem of describing the differences between two images. The output from this algorithm is an image containing the visible differences between the images. Two different visualization techniques are proposed for the output VDP, the free-field difference map optimized for compression, and the in-context difference showing the output probabilities in colour on the reference image.

VDP can be used for all image distortions including blur, noise, algorithm artefacts, banding, blocking, pixilation and tone-scale changes.

B.4.3 Mantiuk, HDR-VDP and HDR-VDP-2

Mantiuk et al. [2004] proposed an extension of VDP for HDR images. The performed extension improves the model's prediction of perceivable differences in the full visible range of luminance and under the adaptation condition. HDR-VDP takes into account aspects of high contrast vision in order to predict perceived differences. This model does not take into account chromatic changes, only luminance.

HDR-VDP-2 [1] is a full reference (FR) image quality metric using a model of human visual system (HVS) to quantify the difference between original and test image. Its main advantage, compared to other FR measures, is the ability to compare images with both standard dynamic range (SDR) and high dynamic range (HDR).

The model includes simulation of optical retinal pathway with intra-ocular light scatter, photoreceptor spectral sensitivity, luminance masking, and achromatic response followed by multi-scale decomposition and neural noise model containing neural contrast sensitivity function (CSF) and contrast masking. It requires setting of several parameters about display and viewing conditions.

The result is a visibility map showing the probabilities of difference detection. This could be pooled to provide a single difference visibility value or quality mean opinion score (MOS). The newest version of the metric – HDR-VDP-2.2 [2] – is calibrated on the representative set of SDR and HDR images to provide more reliable MOS estimations.

B.5 Visual discrimination metrics

B.5.1 VDM

Lubin [1995] proposed the visual discrimination model (VDM). This model is based on the just-noticeable-differences (JND) model by Carlson and Cohen [1980]. The model design was motivated by speed and accuracy. Input to the model is a reference image and a distorted version, both grayscale. A set of parameters must be defined based on the viewing conditions. The first step includes a simulation of the optics of the eye before sampling the retina cone mosaic. The sampling is done by a Gaussian convolution and point sampling. The next stage converts the raw luminance signal into units of local contrast based on a method similar to Peli [1990]. After this each pyramid level is convolved with 4 pairs of spatially oriented filters. Then on the 4 pairs of filters an energy response is computed. Each energy measure is normalized and each of these values are as input to a non-linear sigmoid function. The distance between the vectors can be calculated and results in a JND map as output, but the values across this map can be used to calculate mean, maximum or other statistical measure of the similarity between the images. This single value can further be converted into probability values.

B.5.2 Sarnoff JND Vision Model

The Sarnoff JND Vision Model [Lubin, 1997] is a method of predicting the perceptual ratings that observers will assign to a degraded colour-image sequence relative to its non-degraded counterpart. The model takes two images, an original and a degraded image, and produces an estimate of the perceptual difference between them. The model does a front-end processing to transform the input signals to light outputs (YCbCr to YUV), and then the light output is transformed to psychophysically defined quantities that separately characterize luma and chroma. A luma JND map and a chroma JND map are created. The JND maps are then used for a correlation summary, resulting in a measure of difference between the original and the degraded image. It should be noted that the metric was developed as a video quality metric, showing a high correlation between predicted quality and perceived quality. The model has also been tested on JPEG data, where a high correlation also was found. Lubin concludes that the model has wide applicability as an objective image quality measurement tool.

The Sarnoff JND vision model was submitted for stanndardization to ANSI and as a contribution to the IEEE G-2.1.6 Compression and Processing Subcommittee in 1997. The committee took no action in publishing the model as a standard.

B.6 Colour model differences

B.6.1 S-CIELAB

Zhang and Wandell [1996] proposed a spatial extension to the CIELAB colour metric (Figure B.2). This metric should fulfill two goals, a spatial filtering to simulate the blurring of the HVS and a consistency with the basic CIELAB calculation for large uniform areas. The image is separated into an opponent-colour space, and each opponent colour image is convolved with a kernel determined by the visual spatial sensitivity of that colour dimension. Finally the filtered image is transformed into CIE-XYZ, and this representation is transformed using the CIELAB formulae.

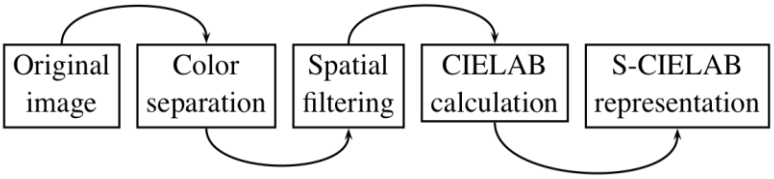

Figure B.2 —Flowchart of the s-CIELAB metric

Colour difference models and mapping has been found useful for identifying areas in an image where coding systems may induce colour shifts or errors that may be visible. Correlation of mapped errors to something visual discernible should be determined with complementary subjective testing.

Annex C

Computational metrics

C.1 Instruction Benchmark

C.1.1 Definition

Count the number of load/store/arithmetic instructions used in a codec. *Time critical or power critical applications, the number of operations of a sequential software version of the CODEC for each class of operations: it is measured by the number of assembler instructions for different classes of instructions of a reference compiler. (Classes of instructions: arithmetic instructions, load-store instructions, branch instructions, remaining instructions)*

C.1.2 C coding benchmarks

This Annex defines several benchmark procedures to measure the performance of image compression algorithms; in addition to the actual measurement itself, each benchmark requires additional deliverables from the metrics defined in Annex B. They are listed in table B-1 and again defined in the corresponding sub-clause.

Table C-1: Deliverables for each Benchmark

Benchmark	Purpose of the test	Primary Deliverable	Secondary Deliverables
Execution Time Benchmark	Estimation of algorithm time complexity under ideal conditions	Execution time per pixel	Hardware specifications, Image Dimensions, PSNR and compression rate
Memory Benchmark	Estimation of space complexity under ideal conditions	Required memory at encoding and decoding time	Hardware specifications, Image Dimensions, PSNR and compression rate
Execution Time vs. Memory Requirement		Execution time per memory unit	Hardware specifications, Image Dimensions, PSNR and compression rate
Cache Hit Rate Benchmark	Performance estimation of data locality	Cache hit rate	Hardware specifications, Image Dimensions, PSNR and compression rate
Parallel Speedup Benchmark		Speedup, Serial Speedup, Efficiency, Throughput	Hardware Specifications, Image Compression, PSNR and compression rate
Bitrate Variation		Bitrate Variation	PSNR and compression rate
Iterative Compression Loss		Average Drift and Average PSNR Loss	

Benchmark	Purpose of the test	Primary Deliverable	Secondary Deliverables
Power consumption		Refer to JTC1 Green ICT	

C.2 Execution-Time Benchmark

C.2.1 Definition

Execution time is here defined in terms of a benchmark process that allows the fair comparison of several codec implementations with respect to a given architecture and given source data. It is defined as the average ratio of time per pixel when a codec encodes or decodes a given source. While other definitions of complexity stress either the asymptotic number of operations for source sizes going to infinity, or the number of arithmetic operations per pixel, it was deemed that such definitions ignore the overhead of memory transfers and cache locality, as well as the ability to utilize architectures like SIMD found on many modern computer architectures. Readers, however, should understand that the guidelines defined here are only suitable to compare two software implementations on the same hardware running under the same conditions including codec settings, and other definitions of complexity are required for hardware implementations. Such measures are beyond this clause.

C.2.2 Measurement Procedure

Procedures to measure the execution times are required by several implementations, measured as a ratio of time per pixel. Use good practices below to ensure fair and reproducible results:

1) The implementations to be compared should be compiled with full optimization(s) enabled; support for profiling or debugging, if applicable, should be disabled.

2) For benchmarking image compression, the implementations should use the same source data set; a standard set of images is provided by WG1 that could be utilized.

3) Choose options of the implementations under investigation such that the execution speed is maximized by ignoring memory requirements and other constraints. Disable execution on multiple cores and/or additional hardware until the test on computational parallelism.

4) For benchmarking decompression, the data source depends on whether benchmarking within standards or across standards is conceived:

5) For benchmarking within the same standard, measure decompressor performance on the same set of bitstreams preferably using the reference implementation of a standard.

6) For benchmarking across standards, test each decompressor on the output of its corresponding compressor.

   i) Within practical limits, measure compressors and decompressors on identical hardware.

   ii) Software benchmarks should use similar computer configurations to the extent possible in terms of CPU, RAM, disk drive type (HDD or SSD).

7) Many modern computer architectures allow adjustable CPU speed, in particular in portable computers. For the purpose of this test, disable such speed adjustments in order to enhance reproducibility of the test.

i) If CPU throttling can be be disabled, a different strategy to ensure maximal CPU speed is to run compression or decompression over several cycles, monitoring the CPU speed and starting the measurement as soon as the operating system increased the CPU clock speed to a maximum. Often, five to ten cycles on the same data are enough to reach maximum performance.

ii) Execution time of the software should be measured over N cycles ignoring results for the first M < N cycles. M should be large enough to ensure that the CPU is clocked at maximal speed and source data is loaded into memory and partially cached in memory. N should be selected large enough to ensure stable results within the measurement precision of the system.

Typical values for N and M are 5 and 25, respectively, but such values may depend on the nature of the source data, of the algorithm; initial tests carefully observing the measurement results must be performed to select reasonable values.

8) Starting with the M+1st cycle, collect the following data:

9) The total running time t_r of the compressor or decompressor for a cycle. This is the end-to-end execution time of the software, not including the time required to load the software into memory, but including the time to load the source data, and including the time to write the output back.

10) The total I/O time t_i required to load source data into the algorithm, and to write output back.

Measuring t_r and t_i typically requires a modification of the software under test. These times can be gathered by using high-precision timers of the operating system or the host CPU. POSIX.1-2001 defines, for example, a function named gettimeofday() that would provide the necessary functionality to implement such time measurements. It should furthermore be noted that N, the total number of cycles, should be large enough to ensure suitable precision.

11) Repeat measurements for defined bitrates to be agreed prior to testing.

12) For compressor performance, read the overall file size S_o for each target bitrate selected.

The result of the benchmark is the average number of milliseconds per megapixel spend for compressing or decompressing an image. It is defined as follows:

$$T := \frac{t_r - t_i}{(N - M) \cdot \sum_{c=0}^{d-1} w(c) \cdot h(c)} \qquad \textbf{Eq. 9}$$

where t_r and t_i are the overall execution time of the program respectively of the I/O operations measured in milliseconds, N is the total number of cycles, M is the number of initial cycles, w is the width of the image in pixels, h the height of the image in pixels and d the number of components (channels) of the image.

Report T, the compression rate and the PSNR for each implementation benchmarked. Report all values for each target bitrate, along with the information on the CPU model, its clock speed and its cache size.

C.3 Memory benchmarking

C.3.1 Definitions

This annex describes the measurement procedures benchmarking the memory requirements of several implementations. As such, the memory requirements are always specific to implementations and not to algorithms, and the purpose of this benchmark is only to compare two or more implementations side by side.

Implementations may offer various modes for compression and decompression of images; this benchmark is designed to measure the peak memory requirement for a compressor or decompressor mode minimizing the required memory. It thus identifies the minimal environment under which an implementation is able to operate.

For example, it is beneficial for this benchmark if an implementation is able to provide a sliding window mode by which only segments of the source image need to be loaded in memory and a continuous stream of output data is generated. Similarly, it is beneficial for a decompressor if it need not to hold the complete image or compressed stream in memory at once and can decompress the image incrementally. It depends, however, also on the codestream design whether such modes are possible, and it is the aim of this benchmark to identify such shortcomings.

It should be understood that algorithms may not perform optimally under memory constrained conditions, and compression performance in terms of execution time or quality may suffer.

C.3.2 Measurement Procedure

This annex defines measurement procedures to measure memory requirements of two implementations, measured in bytes per pixel.

1) For the purpose of this test, repeat measures on the same source data set.

2) Select all options of the two implementations under investigation such that the memory requirements are minimized, ignoring the execution speed and other constraints.

3) For benchmarking decompression, the data source depends on whether benchmarking within standards or across standards is conceived:

    i) Measure decompression performance on the same set of bitstreams/file formats generated preferably by a reference implementation of a standard.

    ii) Test each decompression on the output of its corresponding compression.

1) Measure performance identical hardware architectures, as much as is practical.

2) Monitor the memory allocated by the codec under test continuously. The data to be measured is the maximum amount of memory, measured in bytes, allocated by the codec at a time.

    Measuring the amount of allocated memory may require installing a patch into the software under inspection. A possible strategy for collecting this data might be to replace *malloc()/free()* and/or *operator new/operator delete* by a custom implementation performing the following steps:

    i) Two global variables B and B_m are initialized to zero at program start.

    ii) For each allocation of N bytes, N is stored along with the allocated memory block and B is incremented by N. If B becomes greater than B_m, B_m is set to B.

    iii) Whenever a memory block is released, N is extracted from the memory block and B is decremented by N.

    iv) Other mechanisms for memory allocation may be possible (such as allocating memory from the stack or pre-allocating static memory) and should be included.

    v) On program termination, B_n holds the peak memory requirement to be reported.

3) Repeat measurements for increasing image sizes. It is suggested to approximately double the image size until compression or decompression fails.

4) Repeat measurements for agreed target bitrates within the framework of a core experiment.

5) **Measure compressor performance and record the overall file size S_o for each target bitrate selected.**

   In memory-constraint compression, output rate and target rate might differ significantly. The purpose of this measurement is to estimate the precision up to which a compressor can operate in memory-constraint mode.

6) Record the mean square error combined compressor/decompressor benchmark.

   The purpose of this measurement is not to define an image quality. A separated benchmark exists for this purpose. It is rather designed to identify pathological cases where incorrect or unreasonable compressed streams are generated.

The result of the benchmark is the peak number of bytes per pixel spend for compressing or decompressing the test images. It is defined as follows:

$$A_m = \frac{B_m}{\sum_{c=0}^{d-1} w(c) \cdot h(c)} \qquad \textbf{Eq. 10}$$

where B_m is the peak memory requirement by the codec measured in bytes, $w(c)$ is the width of the channel c in pixels, $h(c)$ the height of channel c and d the number of components (channels) of the image.

Report the values A_m, *bpp* and *PSNR* for each implementation benchmarked and for each target bitrate and for each image size along with the compression rate or bpp, and the image dimensions.

C.4 Cache Hit Rate

C.4.1 Definition

Cache hit rate is defined by the average number of cache hits compared to the total number of memory accesses. An access of the CPU to memory is said to be a *cache hit* if a cache can provide the requested data. If the CPU has to fetch the data from memory or an alternative higher level cache, this access is called a *cache miss*. A codec having a high cache hit rate performs accesses in patterns well-predicted by the CPU cache. It will typically also perform faster than a comparable code having a smaller cache hit rate.

Cache locality is architecture and implementation specific, both need to be reported in the test results. The purpose of this test is, hence, not an absolute measure, but the fair comparison between implementations.

CPUs have typically more than one cache: A relatively small first-level cache, and a second, potentially even third level cache that buffers accesses on first-level cache misses. More than two cache levels might be available as well. Cache locality is mostly interesting for the first-level cache, but results are requested for all available caches of a given CPU architecture.

Measuring cache locality requires a tool that has either direct access to the CPU cache statistics, or implements a simulation of the CPU in software and measures the cache hit rate within this simulation. While WG1 does not provide such a tool, Open Source implementations exist that provide the required functionality, e.g. *valgrind* with its *cachegrind* front-end implements a software simulation that is suitable for the tests outlined here.

C.4.2 Measurement Procedure

This subclause defines measurement procedures to measure cache locality of two implementations, measured in percent of cache hits compared to the total cache access ratio.

7) Perform testing on the same source data set.

8) Measure options of the two implementations under investigation selected to maximize the cache locality and apply comparable options for each coding system under test.

   Selection of proper coding modes is under discretion of the operator of the test, though should be done under a best-effort basis. A couple of pre-tests are suggested to identify coding modes that maximize the cache-hit rate. Typically, these modes are similar to the modes that minimize the memory requirements, see B.2.2.

9) For benchmarking decompression, the data source depends on whether benchmarking within standards or across standards is conceived:

   i) Measure decompressor performance on the same set of bitstreams/file formats generated preferably by a reference implementation of a standard;

   ii) Measure each decompressor on the output of its corresponding compressor.

10) Perform benchmarking on identical hardware architectures as much as practical.

11) Test on a single CPU core.

12) Monitor the the number of cache accesses C_a and cache hits C_h continuously for all caches available for the CPU architecture.

13) Repeat measurements for various target bitrates agreed within the framework of a core experiment.

14) Measure the overall file size S_o for each target bitrate.

15) Especially in memory-constraint compression, output rate and target rate might differ significantly. The purpose of this measurement is to estimate the precision up to which a compressor can operate in memory-constraint mode.

16) Record the mean squared error combined compressor/decompressor benchmark.

    The purpose of this measurement is not to define an image quality. A separated benchmark exists for this purpose. It is rather designed to identify pathological cases where incorrect or unreasonable compressed streams are generated.

The result of the benchmark is the cache hit ratio C_r in percent defined as follows:

$$C_r = 100\frac{C_h}{C_a}$$

where C_h is the total number of cache hits and C_a is the number of cache accesses. Distinguishing between read and write accesses is not necessary for this test, but if the CPU architecture implements more than one cache, measure cache hit ratios individually. The cache hit ratio for the first level cache is then the most interesting result.

Secondary results of compressor benchmarks are the output bitrate *bpp* and the *PSNR* as defined in Annex B.

Report the values C_r, *bpp* and *PSNR* for each implementation benchmarked and for each target bitrate and for each image size along with the target bitrate, the CPU architecture and the image dimensions.

C.5 Degree of Data Parallelism

C.5.1 Definition

Images consist of a set of data on which operations are executed being identical applied to each element of the data set. If data dependencies enable the parallel execution on subsets of data independently, the codec can be implemented in parallel on multi-core CPUs, GPUs or ICs even if the algorithms of the codecs are purely sequential. Therefore, the usage of data parallelism for the parallelization of the codec is the most convenient and effective way to achieve a high performing parallel implementation. The degree of parallelism is defined as the number of independent subsets of data in the above mentioned sense.

C.5.2 Measurement Procedure

The degree of data parallelism *ddp* is measured as the number of data units that can be encoded or decoded independently of each other. In order to relate the degree of data parallelism to the size of the image, the ratio *rdp* is calculated as *rdp=N/ddp* with N being the total number of pixels of the whole image.

NOTE: N can be computed from the image dimensions (see Annex B) as $\sum_{c=0}^{d-1} w(c) \cdot d(c)$

NOTE: The degree of data parallelism requires deep knowledge on the algorithm in question and cannot, generally, be measured by an automated procedure.

C.6 Parallel Speedup Benchmark for PC Systems

C.6.1 Definition

The parallel speedup is defined as the gain in performance for a specific parallel implementation of an algorithm on a multi-core processor or parallel hardware platform for e.g. embedded applications versus a sequential implementation on the same hardware platform or processor. The hardware platform depends on the target application. The primary measure to determine the speedup is the s wall-clock time. In general, the measured time depends on the image size which should be provided by each measurement. In addition to the image size the compression ratio should also be provided with each measurement. The parallel speedup, efficiency, and throughput values are derived from the primary time measured, to support the interpretation of the results of this parallel speedup benchmark. The measurement procedure is similar to the execution time benchmark.

C.6.2 Measurement Procedure

This procedure measures the execution times required by several implementations, measured in milliseconds per megapixel.

1) Compile the implementations to be compared with full optimization enabled.

2) Perform the test implementations on the same source data set.

   The relation between execution time and image size should be expected nonlinear in nature due to caching and bandwidth effects; an image test dataset suitable for the desired target application should be agreed upon at the time of the definition of the test.

3) Select options of the implementations such that the execution speed is maximized.

4) Use the number of execution units allowed in the core experiment framework.

For each measurement the hardware platform or multi-core processor used has to be reported. This includes the number of execution units and their type, the amount of memory and cache available to these execution units, and the interconnection type.

5) The amount of memory needed is introduced in the Memory Benchmark.

6) For benchmarking decompression, the data source depends on whether benchmarking within standards or across standards is conceived:

   i) Measure decompressor performance on the same set of bitstreams/file formats generated preferably by a reference implementation of a standard

   ii) Measure decompressor performance on the output of its corresponding compressor.

7) Test software at maximum available CPU speed on the hardware.

   Many modern computer architectures implement the possibility to adjust the CPU speed dynamically depending on the workload. For the purpose of this test, such speed adjustments limit the reproducibility of the test and hence should be disabled. Failing that, a different strategy to ensure maximal CPU speed is to run compression or decompression over several cycles, monitoring the CPU speed and starting the measurement as soon as the operating system increased the CPU clock speed to a maximum. Often, five to ten cycles on the same data are enough to reach maximum performance.

8) Measure execution time of the software over N cycles ignoring results for the first M < N cycles. Select M large enough to ensure that the CPU is clocked at maximal speed and source data is loaded into memory and partially cached in memory. Select N large enough to ensure stable results within the measurement precision of the system. This measurement ha to be repeated at least three times reporting the average and the variance of the execution time.

   Typical values for N and M are 5 and 25, respectively, but such values may depend on the nature of the source data of the algorithm.

9) Starting with the $M+1^{st}$ cycle, record the following data:

   i) The total running wall-clock time t_r of the compressor or decompressor for a cycle. This is the end-to-end execution time of the software, not including the time required to load the software into memory, but including the time to load the source data, and including the time to write the output back. Also include the time for waiting for some other unit to complete or a resource to be available.

   ii) The total I/O wall-clock time t_i required to load source data into the algorithm, and to write output back. Do not reflect the time needed for synchronization and communication of the parallel execution units.

   iii) The total wall-clock time t_c for communication and synchronization of the execution units.

   Measuring t_r and t_i typically requires a modification of the software under test. These times can be gathered by using high-precision timers of the operating system or the host CPU. POSIX.1-2001 defines, for example, a function named gettimeofday() that would provide the necessary functionality to implement such time measurements. Select N, the total number of cycles, large enough to ensure suitable precision.

10) Repeat measurements for various target bitrates to be agreed on within the framework of a core experiment.

11) Record the overall file size S_o for each target bitrate selected.

The result of the benchmark is the average number of milliseconds per megapixel spend for compressing or decompressing an image on the chosen number of execution units. It is defined as follows:

$$T(P) := \frac{t_r - t_i}{(N - M) \cdot \sum_{c=0}^{d-1} w(c) \cdot h(c)} \qquad \textbf{Eq. 11}$$

here t_r and t_i are the overall execution time of the program respectively of the I/O operations measured in milliseconds, N is the total number of cycles, M is the number of initial cycles and P is the number of parallel units.

NOTE: The scheduling overhead time t_c is by definition already included in the overall time t_r.

Report the values *T,* the compression rate *R* and *PSNR.* for each implementation benchmarked and for each target bitrate, along with the information on the CPU model, its clock speed, the number of cores and their cache sizes.

Further deliverables are the Relative Speedup *S(P,L)*, the Efficiency *E(P,L)*, and the Throughput. They require performing the measurement steps above with a variable number of computing cores, *P*, deployed for compression or decompression.

1) The speedup is defined as:

   S(P) := T(1) / T(P),

   where *T(P)* is the time needed for the parallel version of the algorithm to complete on *P* CPUs/Nodes.

2) Speedup) unless specified otherwise. If T(1) cannot be obtained due to algorithmic constraints and an estimate for this number has been computed instead (see **NOTE**), results must state the procedure how this estimate has been performed.

   Unlike benchmark C.1, the running time *T(1)* includes unavoidable overhead to allow parallel execution, even though not used in this test.

   On some platform it might not be possible to implement *T(1)*. In this case, the speedup needs to be given using a different reference value than the sequential one. Be aware that this ratio does not scale linearly in most cases.

3) If a sequential version of the same algorithm is available for the same platform, the real speedup is this ratio:

   $S_r := T / T(P)$

   where *T* is measured as defined in C.1.

   The real speedup is defined by the factor that a parallel version on *P* computation units runs faster than the best sequential implementation of this algorithm on the same platform. For the applications where the sequential version is an option the C.1 measurement might be run. In this case, the real speedup calculation can be easily done using already done measurements. If only a parallel version is of interest, there is no need to provide an optimized sequential version of the algorithm in addition.

4) Efficiency is defined as:

   E(P) := S(P) /P.

5) Report the values for multiple combinations of parallel computing cores *P* and multiple image sizes. At least four different values for *P*, including *P=1* should be used.

6) Throughput is defined as the number of pixels compressed or decompressed per time:

$$C(P) := \frac{\sum_{c=0}^{d-1} w(c) \cdot h(c)}{T(P)} \qquad \text{Eq. 12}$$

The output of this benchmark consists of the following information:

1) The time *T(P),*

2) The compression rate (see Annex A)

3) The amount of memory required, as defined by Benchmark C.2

4) The variance of the wall-clock time *T(P)* over several measurement cycles

5) The throughput *C(P).*

6) The relative speedup *S(P).*

7) The efficiency *E(P).*

C.7 Implementation Benchmark for Parallel PC System Utilization (Balancing)

C.7.1 Definition

Parallel system utilization is defined as the degree of utilization of all processors in multi-core processor systems during parallel execution of a process. This value is an indicator how well smaller processes are distributed and executed on parallel execution units. A well balanced workload does not necessarily yield a higher execution speed, but parallel system utilization can be used to describe potential free processing resources.

C.7.2 Measurement Procedure

This sub-clause defines measurement procedures to measure the execution times required by several implementations, measured in milliseconds per megapixel. Apply the following recommendations to ensure the reliability and reproducibility of the data:

1) All CPU/Core/Nodes on the evaluation system have identical computing power, ideally the same processors, i.e. this benchmark is only suitable for symmetric multiprocessing hardware.

2) Compile coding implementations to be compared with full optimization enabled and disable any support for profiling or debugging.

3) For benchmarking image compression, execute the implementations under test on the same source image set.

   NOTE: The relation between execution time and image size should be expected to be nonlinear in nature due to caching and bandwidth effects.

4) Maximize execution speed by carefully selecting options of the implementations under investigation.

5) Usually a core experiment defines the number of execution units allowed to be used for applying this benchmark. If possible, disable execution units above the desired number.

NOTE: For each measurement the hardware platform or multi-core processor used has to be reported. This includes the number of execution units and their type, the amount of memory and cache available to these execution units, and the interconnection type.

6) Obtain the amount of memory needed from the Memory Benchmark test (See Section C.3).

7) For benchmarking decompression, the data source depends on whether benchmarking within standards or across standards is conceived:

   i) For benchmarking within the same standard, measure the decoder performance on identical bitstreams. In general, a coding system's reference bitstream should be used.

   ii) For benchmarking across standards, test each decoder on the output of its corresponding encoder.

8) Execute software at the maximum available CPU speed of the available hardware and record the speed.

   NOTE: Many modern computer architectures implement the possibility to adjust the CPU speed dynamically depending on the workload. For the purpose of this test, such speed adjustments limit the reproducibility of the test and hence should be disabled. Failing that, a different strategy to ensure maximal CPU speed is to run compression or decompression over several cycles, monitoring the CPU speed and starting the measurement as soon as the operating system increased the CPU clock speed to a maximum. Often, five to ten cycles on the same data are enough to reach maximum performance.

9) Execution time of the software shall be measured over N cycles ignoring results for the first M < N cycles. M shall be selected large enough to ensure that the CPU is clocked at maximal speed and source data is loaded into memory and partially cached in memory. N shall be selected large enough to ensure stable results within the measurement precision of the system. This measurement has to be repeated at least 3 times reporting the average and the variance of the execution time.

   NOTE: Typical values for N and M are 5 and 25, respectively, but such values may depend on the nature of the source data, of the algorithm; initial tests carefully observing the measurement results should be performed to select reasonable values.

10) Collect the following data, starting with the M+1st cycle:

11) The total running wall-clock time t_r of the compressor or decompressor for a cycle. This is the end-to-end execution time of the software, **not** including the time required to load the software into memory, **but** including the time to load the source data, **and** including the time to write the output back. Also the time for waiting for some other unit to complete or a resource to be available shall be included here.

12) The total I/O wall-clock time t_i required to load source data into the algorithm, and to write output back. The wall clock time should not include the time needed for synchronization and communication of the parallel execution units (see next step).

13) The total wall-clock time t_c for communication and synchronization of the execution units.

    NOTE: Measuring t_r and t_i typically requires a modification of the software under test. These times can be gathered by using high-precision timers of the operating system or the host CPU. POSIX.1-2001 defines, for example, a function named gettimeofday() that would provide the necessary functionality to implement such time measurements. Collect a sufficient number of cycles, N, large enough to ensure suitable precision.

14) Repeat measurements for the target bitrates usually listed in the framework of a core experiment. For example, test at 0.25, 0.5, 0.75, 1.0, 1.5, 2.0 BPP. Testing may include the lossless coding performance.

15) Record the overall file size S_o for each target bitrate selected.

The result of the benchmark is the average parallel system utilization for compressing or decompressing an image on the chosen number of execution units. It is defined as follows:

$$t_p(A) := \sum_{a=0}^{A-1} t_{process}(a) \qquad \textbf{Eq. 13}$$

A is the number execution unit state changes indicating the switches from idle state to processing state. $t_{process}$ (a) defines the uninterrupted period in which the *a*-th execution unit stays in processing state. t_a (P) defines therefore the overall processing time assigned to the *p*-th execution unit.

NOTE: The scheduling overhead time t_c is by definition already included in $t_a(P)$.

$$t_{active}(P) := \sum_{p=0}^{P-1} t_p \qquad \textbf{Eq. 14}$$

P is the total number of considered parallel execution units. $t_{active}(P)$ is therefore the overall time all execution units staying in processing state.

The parallel system utilization is defined as:

$$U(P) := \frac{t_{active}(P)}{t_r - t_i} \qquad \frac{t_r - t_i}{A} \le U(A) \le 1 \qquad \textbf{Eq. 15}$$

If *U(A)* is equal 1 the job could or has been done in sequential order.

The above indicators require performing the measurement steps above with a variable number of computing cores, *P*, deployed for compression or decompression. Report the following:

1) The system utilization *U(P),*

2) The compression rate (see Annex A)

3) The amount of memory required, as defined by Benchmark C.2

4) The throughput *C(P).*

Annex D
(informative)

Verification of codec charactistics

D.1 Variable bit rate variation

This measurement procedure determines the maximum bitrate variation of an image compression codec. To enable such a test, the compression codec under evaluation must provide a mode of operation under which an input image can be feed iteratively in smaller units, here called minimal coding units, into the codec. Depending on the compressor, minimum coding units can be individual pixels, stripes or blocks of image data. Such an operation mode is also called online-capable. If the code is not online-capable, this benchmark cannot be implemented.

If possible, this benchmark should be complemented by a theoretical analysis of the worst-case bitrate variance.

1) If the compressor under inspection offers several compression modes, then online-capable modes and the mode minimizing the bitrate variation can be chosen as appropriate for a fair and reliable test.

2) Report the minimal coding unit of the coding system.

3) Use suitable control images and test images agreed prior to testing. Such a suite of test image data can be obtained from the WG1.

4) For each coding unit U feed into the compressor, the number of image bits contained in this unit equals:

$$B_i = \frac{1}{N} \sum_{c \in U} \sum_{(x,y) \in U} b(c) \qquad \text{Eq. 16}$$

5) where U is the coding unit, c is the image channel and (x,y) are the pixel positions within the image, and the sum runs over all pixels within the coding unit. The quantity b(x,y,c) is the bit precision of the pixel, as defined in Annex D.

6) For each coding unit feed into the compressor, the output bitrate is :

$$B_o = 8 \cdot b$$

7) Where b is the number of bytes leaving the codec.

8) The bitrate for unit U is defined as $R(U):=B_o/B_i$, the number of output bits generated for each input bit.

9) Measure R(U) for each coding unit going into the compression codec until all of the image is compressed, giving a function of the bitrate over the percentage of the completion of the image.

D.2 Generational quality loss

This procedure measures the generation quality loss and the DC drift of a single codec. To this end, compressor and decompressor must be provided. If the compressor/decompressor specification allows certain

freedoms, generation loss also depends on the implementation, and generation loss can be measured against a reference implementation if available.

1) Select a compressor/decompressor pair to measure the generation loss. If a reference implementation is available, the vendor compressor can be measured against the reference decompressor, and the reference compressor measured against the vendor decompressor.

2) Select coding/decoding parameters that minimize generation loss.

3) Repeat the following steps for a selection of test images typical for the application domain of the codec.

4) Repeat the steps for several target bit rates to be agreed within the framework of a core experiment.

5) Set *n* to zero and the image I_0 to the source image.

6) Compress the image I_n to the target bit rate, decompress the resulting stream giving image I_{n+1} and measure the following data for n > 0:

   i) The archived bit rate of the compressor defined as the average number of bits per pixel,

   ii) The $PSNR_n$ between the first generation image I_1 and I_{n+1}.

   iii) The drift error D_c for each channel/component between the image:

$$D_c = \frac{1}{w(c) \cdot h(c)} \sum_{x=0}^{w(c)-1} \sum_{y=0}^{h(c)-1} (p(x, y, c) - q(x, y, c)) \qquad \text{Eq. 17}$$

   where *w* is the width, *h* the height and *d* the number of components/channels of the source image and *p(x,y,c)* is the channel value of the pixel at position *x,y* in channel *c* of the original image, *q(x,y,c)* the channel value at the same location in the distorted image.

7) Increment n and repeat the step above, continuously measuring $PSNR_n$ and $D_{c,n}$ for each iteration. Repeat these steps *N* times, where *N* should be at least five.

8) The result of the test is the average drift D_c for each component *c* per component and the average PNSR-loss per generation, *PSNR* defined as follows:

$$D_c = \frac{1}{N-1} \sum_{n=1}^{N-1} D_{c,n} \quad PNSR = \frac{1}{N-1} \sum_{n=1}^{N-1} PSNR_n \qquad \text{Eq. 18}$$

   where N is the number of iterations (generations) over which the measurement has been performed. The value D_c is called the average drift, the value *PSNR* is the average quality loss.

D.3 Error resiliency

One way to test error resiliency is to inject errors into a stream and view the output result to observe if image artefacts are present either through objective or subjective means

Bibliography

[1] BRYCE E. BAYER; Colour imaging array; United States Patent , March 5, 1975

[2] CIE DS 014-6/E:2012, “Colorimetry - Part 6: CIEDE2000 colour-difference formula,” CIE Central Bureau, Vienna (2012)

[3] D. KOFF, P. BAK, P. BROWNRIGG, D. HOSSEINZADEH, A. KHADEMI, A. KISS, L. LEPANTO, T. MICHALAK, H. SHULMAN, AND A. VOLKENING, "Pan-Canadian evaluation of irreversible compression ratios ("Lossy" compression) for development of national guidelines," J Digit Imag, vol. 22, pp. 569-578, Dec. 2009

[4] ISO 3664, *Graphic technology and photography – Viewing conditions*, ISO/TC 42

[5] ISO 9241-303, *Ergonomics of human-system interaction – Requirements for electronics visual displays*, ISO/TC 159 SC 4

[6] ISO 20462-2, *Photography – Psychophysical experimental method for estimating image quality – Triplet comparison method,* ISO/TC 42

[7] ISO 20462-3, *Photography – Psychophysical experimental method for estimating image quality – Quality ruler method,* ISO/TC 42

[8] ISO/IEC 29170-2, *Information technology – Advanced image coding and evaluation – Evaluation procedure for nearly lossless coding,* ISO/IEC JTC 1/SC 29

[9] J. LUBIN, "A human vision system model for objective picture quality measurements", Broadcasting Convention, 1997. International, September 12, 1997

[10] JOHN F. JR. HAMILTON, JOHN T. COMPTON; Processing colour and panchromatic pixels; United States Patent Application; July 28, 2005

[11] K. J. KIM, B. KIM, S. W. CHOI, Y. H. KIM, S. HAHN, T. J. KIM, S. J. CHA, V. BAJPAI, AND K. H. LEE, "Definition of compression ratio: difference between two commercial JPEG2000 program libraries," Telemed J E Health, vol. 14, pp. 350-354, May, 2008

[12] M. NARWARIA, R. K. MANTIUK, M. PERREIRA DA SILVA, AND P. LE CALLET, HDR-VDP-2.2: a calibrated method for objective quality prediction of high-dynamic range and standard images, J. Electron. Imaging. 24(1), 2015

[13] R. M. SLONE, D. H. FOOS, B. R. WHITING, E. MUKA, D. A. RUBIN, T. K. PILGRAM, K. S. KOHM, S. S. YOUNG, P. HO, AND D. D. HENDRICKSON, "Assessment of visually lossless irreversible image compression: comparison of three methods by using an image-comparison workstation," Radiology, vol. 215, pp. 543-553, May, 2000

[14] R. MANTIUK, K. J. KIM, A. G. REMPEL, AND W. HEIDRICH. HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. In ACM Transactions on Graphics (Proc. of SIGGRAPH’11), volume 30, 2011

[15] Recommendation ITU-R BT.500, *Methodology for the subjective assessment of the quality of television pictures* ITU-R SG 6

[16] Recommendation ITU-T J.340, *Cable networks and transmission of television, sound programme and other multimedia signals – Measure of the quality of service*, ITU-T SG 9

[17] Recommendation ITU-T P.910, *Telephone transmission quality, telephone installations, local line networks – Audiovisual quality in multimedia services,* ITU-T SG 9