Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 09, 2017

T. Daede
Mozilla
A. Norkin
Netflix
I. Brailovskiy
Amazon Lab126

July 08, 2016

Video Codec Testing and Quality Measurement
draft-ietf-netvc-testing-03

Abstract

This document describes guidelines and procedures for evaluating a video codec. This covers subjective and objective tests, test conditions, and materials used for the test.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 09, 2017.

Table of Contents

1. Introduction

When developing a video codec, changes and additions to the codec need to be decided based on their performance tradeoffs. In addition, measurements are needed to determine when the codec has met its performance goals. This document specifies how the tests are to be carried about to ensure valid comparisons when evaluating changes under consideration. Authors of features or changes should provide the results of the appropriate test when proposing codec modifications.

2. Subjective quality tests

Subjective testing is the preferable method of testing video codecs.

Subjective testing results take priority over objective testing results, when available. Subjective testing is recommended especially when taking advantage of psychovisual effects that may not be well represented by objective metrics, or when different objective metrics disagree.

Selection of a testing methodology depends on the feature being tested and the resources available. Test methodologies are presented in order of increasing accuracy and cost.

Testing relies on the resources of participants. For this reason, even if the group agrees that a particular test is important, if no one volunteers to do it, or if volunteers do not complete it in a timely fashion, then that test should be discarded. This ensures that only important tests be done in particular, the tests that are important to participants.

2.1. Still Image Pair Comparison

A simple way to determine superiority of one compressed image is to visually compare two compressed images, and have the viewer judge which one has a higher quality. This is used for rapid comparisons during development - the viewer may be a developer or user, for example. Because testing is done on still images (keyframes), this is only suitable for changes with similar or no effect on other frames. For example, this test may be suitable for an intra de-ringing filter, but not for a new inter prediction mode. For this test, the two compressed images should have similar compressed file sizes, with one image being no more than 5% larger than the other. In addition, at least 5 different images should be compared.

2.2. Video Pair Comparison

Video comparisons are necessary when making changes with temporal effects, such as changes to inter-frame prediction. Video pair comparisons follow the same procedure as still images.

2.3. Subjective viewing test

A subjective viewing test is the preferred method of evaluating the quality. The subjective test should be performed as either consecutively showing the video sequences on one screen or on two screens located side-by-side. The testing procedure should normally follow rules described in [BT500] and be performed with non-expert test subjects. The result of the test could be (depending on the test procedure) mean opinion scores (MOS) or differential mean opinion scores (DMOS). Normally, confidence intervals are also calculated to judge whether the difference between two encodings is statistically significant. In certain cases, a viewing test with expert test subjects can be performed, for example if a test should evaluate technologies with similar performance with respect to a particular artifact (e.g. loop filters or motion prediction). Depending on the setup of the test, the output could be a MOS, DMOS or a percentage of experts, who preferred one or another technology.

3. Objective Metrics

Objective metrics are used in place of subjective metrics for easy and repeatable experiments. Most objective metrics have been designed to correlate with subjective scores.

The following descriptions give an overview of the operation of each of the metrics. Because implementation details can sometimes vary, the exact implementation is specified in C in the Daala tools repository [DAALA-GIT]. Implementations of metrics must directly support the input's resolution, bit depth, and sampling format.

Unless otherwise specified, all of the metrics described below only apply to the luma plane, individually by frame. When applied to the video, the scores of each frame are averaged to create the final score.

Codecs must output the same resolution, bit depth, and sampling format as the input.

3.1. Overall PSNR

PSNR is a traditional signal quality metric, measured in decibels. It is directly drived from mean square error (MSE), or its square root (RMSE). The formula used is:

20 * log10 (MAX / RMSE)

or, equivalently:

10 * log10 (MAX^2 / MSE)

where the error is computed over all the pixels in the video, which is the method used in the dump_psnr.c reference implementation.

This metric may be applied to both the luma and chroma planes, with all planes reported separately.

3.2. Frame-averaged PSNR

PSNR can also be calculated per-frame, and then the values averaged together. This is reported in the same way as overall PSNR.

3.3. PSNR-HVS-M

The PSNR-HVS metric performs a DCT transform of 8x8 blocks of the image, weights the coefficients, and then calculates the PSNR of those coefficients. Several different sets of weights have been considered. [PSNRHVS] The weights used by the dump_pnsrhvs.c tool in the Daala repository have been found to be the best match to real MOS scores.

3.4. SSIM

SSIM (Structural Similarity Image Metric) is a still image quality metric introduced in 2004 [SSIM]. It computes a score for each individual pixel, using a window of neighboring pixels. These scores can then be averaged to produce a global score for the entire image. The original paper produces scores ranging between 0 and 1.

For the metric to appear more linear on BD-rate curves, the score is converted into a nonlinear decibel scale:

-10 * log10 (1 - SSIM)

3.5. Multi-Scale SSIM

Multi-Scale SSIM is SSIM extended to multiple window sizes [MSSSIM].

3.6. Fast Multi-Scale SSIM

Fast MS-SSIM is a modified implementation of MS-SSIM which operates on a limited number of scales and with modified weights [FASTSSIM]. The final score is converted to decibels in the same manner as SSIM.

3.7. CIEDE2000

CIEDE2000 is a metric based on CIEDE color distances [CIEDE2000]. It generates a single score taking into account all three chroma planes. It does not take into consideration any structural similarity or other psychovisual effects.

3.8. VMAF

Video Multi-method Assessment Fusion (VMAF) is a full-reference perceptual video quality metric that aims to approximate human perception of video quality [VMAF]. This metric is focused on quality degradation due compression and rescaling. VMAF estimates the perceived quality score by computing scores from multiple quality assessment algorithms, and fusing them using a support vector machine (SVM). Currently, three image fidelity metrics and one temporal signal have been chosen as features to the SVM, namely Anti-noise SNR (ANSNR), Detail Loss Measure (DLM), Visual Information Fidelity (VIF), and the mean co-located pixel difference of a frame with respect to the previous frame.

4. Comparing and Interpreting Results

4.1. Graphing

When displayed on a graph, bitrate is shown on the X axis, and the quality metric is on the Y axis. For publication, the X axis should be linear. The Y axis metric should be plotted in decibels. If the quality metric does not natively report quality in decibels, it should be converted as described in the previous section.

4.2. BD-Rate

The Bjontegaard rate difference, also known as BD-rate, allows the measurement of the bitrate reduction offered by a codec or codec feature, while maintaining the same quality as measured by objective metrics. The rate change is computed as the average percent difference in rate over a range of qualities. Metric score ranges are not static - they are calculated either from a range of bitrates of the reference codec, or from quantizers of a third, anchor codec. Given a reference codec and test codec, BD-rate values are calculated as follows:

- Rate/distortion points are calculated for the reference and test codec.
  - At least four points must be computed. These points should be the same quantizers when comparing two versions of the same codec.
  - Additional points outside of the range should be discarded.
- The rates are converted into log-rates.
- A piecewise cubic hermite interpolating polynomial is fit to the points for each codec to produce functions of log-rate in terms of distortion.
- Metric score ranges are computed:
  - If comparing two versions of the same codec, the overlap is the intersection of the two curves, bound by the chosen quantizer points.
  - If comparing dissimilar codecs, a third anchor codec's metric scores at fixed quantizers are used directly as the bounds.
- The log-rate is numerically integrated over the metric range for each curve, using at least 1000 samples and trapezoidal integration.
- The resulting integrated log-rates are converted back into linear rate, and then the percent difference is calculated from the reference to the test codec.

4.3. Ranges

For all tests described in this document, the anchor codec used for ranges is libvpx 1.5.0 run with VP9 and High Latency CQP settings. The quality range used is that achieved between cq-level 20 and 55. For testing changes to libvpx or libaom, the anchor does not need to be used.

5. Test Sequences

5.1. Sources

Lossless test clips are preferred for most tests, because the structure of compression artifacts in already-compressed clips may introduce extra noise in the test results. However, a large amount of content on the internet needs to be recompressed at least once, so

some sources of this nature are useful. The encoder should run at the same bit depth as the original source. In addition, metrics need to support operation at high bit depth. If one or more codecs in a comparison do not support high bit depth, sources need to be converted once before entering the encoder.

5.2. Test Sets

Sources are divided into several categories to test different scenarios the codec will be required to operate in. For easier comparison, all videos in each set should have the same color subsampling, same resolution, and same number of frames. In addition, all test videos must be publicly available for testing use, to allow for reproducibility of results. All current test sets are available for download [TESTSEQUENCES].

Test sequences should be downloaded in whole. They should not be recreated from the original sources.

5.2.1. regression-1

This test set is used for basic regression testing. It contains a very small number of clips.

- kirlandvga (640x360, 8bit, 4:2:0, 300 frames)
- FourPeople (1280x720, 8bit, 4:2:0, 60 frames)
- Narrarator (4096x2160, 10bit, 4:2:0, 15 frames)
- CSGO (1920x1080, 8bit, 4:4:4 60 frames)

5.2.2. objective-1

This test set is a comprehensive test set, grouped by resolution. These test clips were created from originals at [TESTSEQUENCES]. They have been scaled and cropped to match the resolution of their category. Other deviations are noted in parenthesis.

4096x2160, 10bit, 4:2:0, 60 frames:

- Aerial (start frame 600)
- BarScene (start frame 120)
- Boat (start frame 0)
- BoxingPractice (start frame 0)

- Crosswalk (start frame 0)
- Dancers (start frame 120)
- FoodMarket
- Narrator
- PierSeaside
- RitualDance
- SquareAndTimelapse
- ToddlerFountain (start frame 120)
- TunnelFlag
- WindAndNature (start frame 120)

1920x1080, 8bit, 4:4:4, 60 frames:

- CSGO
- DOTA2
- EuroTruckSimulator2
- Hearthstone
- MINECRAFT
- STARCRAFT
- wikipedia
- pvq_slideshow

1920x1080, 8bit, 4:2:0, 60 frames:

- ducks_take_off
- life
- aspen
- crowd_run

- old_town_cross
- park_joy
- pedestrian_area
- rush_field_cuts
- rush_hour
- station2
- touchdown_pass

1280x720, 8bit, 4:2:0, 60 frames:

- Netflix_FoodMarket2
- Netflix_Tango
- DrivingPOV (start frame 120)
- DinnerScene (start frame 120)
- RollerCoaster (start frame 600)
- FourPeople
- Johnny
- KristenAndSara
- vidyo1
- vidyo3
- vidyo4
- dark720p
- gipsrecmotion720p
- gipsrestat720p
- controlled_burn
- stockholm

- speed_bag
- snow_mnt
- shields

640x360, 8bit, 4:2:0, 60 frames:

- red_kayak
- blue_sky
- riverbed
- thaloundeskmtgvga
- kirlandvga
- tacomanarrowsvga
- tacomascmvvga
- desktop2360p
- mmmovingvga
- mmstationaryvga
- niklasvga

5.2.3. objective-1-fast

This test set is based on objective-1, but requires much less computation. It is intended to be a predictor for the results from objective-1.

2048x1080, 8bit, 4:2:0, 60 frames:

- Aerial (start frame 600)
- Boat (start frame 0)
- Crosswalk (start frame 0)
- FoodMarket
- PierSeaside

- SquareAndTimelapse
- TunnelFlag

1920x1080, 8bit, 4:2:0, 60 frames:

- CSGO
- EuroTruckSimulator2
- MINECRAFT
- wikipedia

1920x1080, 8bit, 4:2:0, 60 frames:

- ducks_take_off
- aspen
- old_town_cross
- pedestrian_area
- rush_hour
- touchdown_pass

1280x720, 8bit, 4:2:0, 60 frames:

- Netflix_FoodMarket2
- DrivingPOV (start frame 120)
- RollerCoaster (start frame 600)
- Johnny
- vidyo1
- vidyo4
- gipsrecmotion720p
- speed_bag
- shields

640x360, 8bit, 4:2:0, 60 frames:

- red_kayak
- riverbed
- kirlandvga
- tacomascmvvga
- mmmovingvga
- niklasvga

5.3. Operating Points

Four operating modes are defined. High latency is intended for on demand streaming, one-to-many live streaming, and stored video. Low latency is intended for videoconferencing and remote access. Both of these modes come in CQP and unconstrained variants. When testing still image sets, such as subset1, high latency CQP mode should be used.

5.3.1. Common settings

Encoders should be configured to their best settings when being compared against each other:

- av1: -codec=av1 -ivf -frame-parallel=0 -tile-columns=0 -cpu-used=0 -threads=1

5.3.2. High Latency CQP

High Latency CQP is used for evaluating incremental changes to a codec. This method is well suited to compare codecs with similar coding tools. It allows codec features with intrinsic frame delay.

- daala: -v=x -b 2
- vp9: -end-usage=q -cq-level=x -lag-in-frames=25 -auto-alt-ref=2
- av1: -end-usage=q -cq-level=x -lag-in-frames=25 -auto-alt-ref=2

5.3.3. Low Latency CQP

Low Latency CQP is used for evaluating incremental changes to a codec. This method is well suited to compare codecs with similar coding tools. It requires the codec to be set for zero intrinsic frame delay.

- daala: -v=x
- av1: -end-usage=q -cq-level=x -lag-in-frames=0

5.3.4. Unconstrained High Latency

The encoder should be run at the best quality mode available, using the mode that will provide the best quality per bitrate (VBR or constant quality mode). Lookahead and/or two-pass are allowed, if supported. One parameter is provided to adjust bitrate, but the units are arbitrary. Example configurations follow:

- x264: -crf=x
- x265: -crf=x
- daala: -v=x -b 2
- av1: -end-usage=q -cq-level=x -lag-in-frames=25 -auto-alt-ref=2

5.3.5. Unconstrained Low Latency

The encoder should be run at the best quality mode available, using the mode that will provide the best quality per bitrate (VBR or constant quality mode), but no frame delay, buffering, or lookahead is allowed. One parameter is provided to adjust bitrate, but the units are arbitrary. Example configurations follow:

- x264: -crf-x -tune zerolatency
- x265: -crf=x -tune zerolatency
- daala: -v=x
- av1: -end-usage=q -cq-level=x -lag-in-frames=0

6. Automation

Frequent objective comparisons are extremely beneficial while developing a new codec. Several tools exist in order to automate the process of objective comparisons. The Compare-Codecs tool allows BD-rate curves to be generated for a wide variety of codecs [COMPARECODECS]. The Daala source repository contains a set of scripts that can be used to automate the various metrics used. In addition, these scripts can be run automatically utilizing distributed computers for fast results, with rd_tool [RD_TOOL]. This tool can be run via a web interface called AreWeCompressedYet [AWCY], or locally.

Because of computational constraints, several levels of testing are specified.

6.1. Regression tests

Regression tests run on a small number of short sequences - regression-test-1. The regression tests should include a number of various test conditions. The purpose of regression tests is to ensure bug fixes (and similar patches) do not negatively affect the performance. The anchor in regression tests is the previous revision of the codec in source control. Regression tests are run on both high and low latency CQP modes

6.2. Objective performance tests

Changes that are expected to affect the quality of encode or bitstream should run an objective performance test. The performance tests should be run on a wider number of sequences. The following data should be reported:

- Identifying information for the encoder used, such as the git commit hash.
- Command line options to the encoder, configure script, and anything else necessary to replicate the experiment.
- The name of the test set run (objective-1)
- For both high and low latency CQP modes, and for each objective metric:
  - The BD-Rate score, in percent, for each clip.
  - The average of all BD-Rate scores, equally weighted, for each resolution category in the test set.

* The average of all BD-Rate scores for all videos in all categories.

For non-tool contributions, the test set objective-1-fast can be substituted.

6.3. Periodic tests

Periodic tests are run on a wide range of bitrates in order to gauge progress over time, as well as detect potential regressions missed by other tests.

7. Informative References

[AWCY] Xiph.Org, "Are We Compressed Yet?", 2016, <https://arewecompressedyet.com/>.

[BT500] ITU-R, "Recommendation ITU-R BT.500-13", 2012, <https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.500-13-201201-I!!PDF-E.pdf>.

[CIEDE2000] Yang, Y., Ming, J., and N. Yu, "Color Image Quality Assessment Based on CIEDE2000", 2012, <http://dx.doi.org/10.1155/2012/273723>.

[COMPARECODECS] Alvestrand, H., "Compare Codecs", 2015, <http://compare-codecs.appspot.com/>.

[DAALA-GIT] Xiph.Org, "Daala Git Repository", 2015, <http://git.xiph.org/?p=daala.git;a=summary>.

[DERFVIDEO] Terriberry, T., "Xiph.org Video Test Media", n.d., <https://media.xiph.org/video/derf/>.

[FASTSSIM] Chen, M. and A. Bovik, "Fast structural similarity index algorithm", 2010, <http://live.ece.utexas.edu/publications/2011/chen_rtip_2011.pdf>.

[L1100] Bossen, F., "Common test conditions and software reference configurations", JCTVC L1100, 2013, <http://phenix.int-evry.fr/jct/>.

[MSSSIM] Wang, Z., Simoncelli, E., and A. Bovik, "Multi-Scale Structural Similarity for Image Quality Assessment", n.d., <http://www.cns.nyu.edu/~zwang/files/papers/msssim.pdf>.

[PSNRHVS] Egiazarian, K., Astola, J., Ponomarenko, N., Lukin, V., Battisti, F., and M. Carli, "A New Full-Reference Quality Metrics Based on HVS", 2002.

[RD_TOOL] Xiph.Org, "rd_tool", 2016, <https://github.com/tdaede/rd_tool>.

[SSIM] Wang, Z., Bovik, A., Sheikh, H., and E. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity", 2004, <http://www.cns.nyu.edu/pub/eero/wang03-reprint.pdf>.

[STEAM] Valve Corporation, "Steam Hardware & Software Survey: June 2015", June 2015, <http://store.steampowered.com/hwsurvey>.

[TESTSEQUENCES] Daede, T., "Test Sets", n.d., <https://people.xiph.org/~tdaede/sets/>.

[VMAF] Aaron, A., Li, Z., Manohara, M., Lin, J., Wu, E., and C. Kuo, "VMAF - Video Multi-Method Assessment Fusion", 2015, <https://github.com/Netflix/vmaf>.

Authors' Addresses

Thomas Daede
Mozilla

Email: tdaede@mozilla.com

Andrey Norkin
Netflix

Email: anorkin@netflix.com

Ilya Brailovskiy
Amazon Lab126

Email: brailovs@lab126.com