Reproducible research framework for objective video quality measures using a large-scale database approach

Ahmed Aldahdooh[1], Enrico Masala[2], Glenn Van Wallendael[3], Marcus Barkowsky[1]

[1] *LUNAM Université, Université de Nantes, IRCCyN UMR CNRS 6597, Nantes, France*
[2] *Control and Computer Engineering Department*
Politecnico di Torino, corso Duca degli Abruzzi 24, 10129 Torino - Italy,
[3] *Ghent University - imec - IDLab, Ghent, Belgium*

Abstract

This work presents a framework to facilitate reproducibility of research in video quality evaluation. Its initial version is built around the JEG-Hybrid database of HEVC coded video sequences. The framework is modular, organized in the form of pipelined activities, which range from the tools needed to generate the whole database from reference signals up to the analysis of the video quality measures already present in the database. Researchers can rerun, modify and extend any module, starting from any point in the pipeline, while always achieving perfect reproducibility of the results. The modularity of the structure allows to work on subsets of the database since for some analysis this might be too computationally intensive. To this purpose, the framework also includes a software module to compute interesting subsets, in terms of coding conditions, of the whole database. An example shows how the framework can be used to investigate how the small differences in the definition of the widespread PSNR metric can yield very different results, discussed in more details in our accompanying research paper [1]. This further underlines the importance of reproducibility to allow comparing different research work with high confidence. To the best of our knowledge, this framework is the first attempt to bring exact reproducibility end-to-end in the context of video quality evaluation research.

Keywords: Reproducible research, large database analysis, video quality

1. Motivation and significance

The domain of objective video quality algorithms suffers from the lack of reproducible research. Scientific progress is impacted by missing implementations of existing algorithms and missing test data. The implementations are often missing because the individual authors hesitate to publish their code, thus requiring reimplementation of complex algorithms. As test data for the correctness of the algorithms is often also missing, a reimplementation may not be validated. As a consequence, comparisons published in the domain rely on uncertain data.

The software described in this paper serves three purposes. Firstly, it provides an environment for calculating a reproducible large-scale dataset of compressed video sequences that can be used both as test data and for comparisons of algorithms. Secondly, the framework provides the possibility to provide different Peak Signal to Noise Ratio (PSNR) measures as an example for the calculation of more complex algorithms and for comparisons in scientific research as most researchers compare their work to PSNR. Thirdly, a subset selection algorithm that can be targeted to different research questions is provided that allows for running computationally expensive algorithms on parts of the large-scale database while retaining important characteristics for the analysis.

This software package helps in developing and evaluating objective video quality measurement algorithms. Researchers can reproduce the test dataset and provide the results of their algorithm so that exact reproducibility of their algorithm can be guaranteed. Comparisons between different algorithms are also enabled because of the size of the large-scale dataset which reduces the probability of overtraining if the test design is carefully chosen. Last but not least, it makes researchers in the field aware that even small differences in algorithms may lead to important differences in the conclusions of their algorithms thus providing motivation for precise descriptions or published reference software.

An example of such a study using this framework is proposed in [1]

2. Software description

The presented software solution provides the glue for a set of available and new tools with video-quality research reproducibility as its biggest goal.

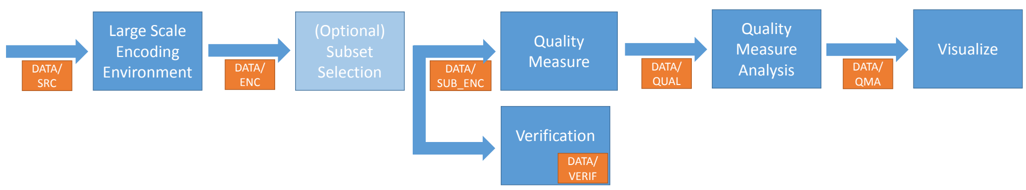

Figure 1: Software architecture showing the pipeline of active components (large blue boxes) and communication directories (small orange boxes).

2.1. Software Architecture

The software consists of a pipeline architecture where each active component communicates to the other component using directories (see Fig. 1). Each individual component will be explained next.

2.1.1. Large Scale Encoding Environment

The "Large Scale Encoding Environment" performs the Hypothetical Reference Circuit (HRC) processing. At this moment, this module consists the version used in this evaluation of the HEVC standardization reference encoding package (http://hevc.kw.bbc.co.uk/svn/jctvc-a124/tags/HM-11.1/) accompanied by a valuable set of configuration files and scripts in order to reproduce or extend the first version of the HEVC large scale database. Whenever the proposed video quality analysis framework needs improvement with more recent versions, more compression standards, other compression parameters, or network impairment simulations, then solely this block must be extended upon. Please note that more recent versions of the HEVC reference software may change the results of this reproducible dataset and should thus be considered a different dataset.

2.1.2. Subset Selection

This process facilitates the research work if proper HRCs are selected to represent a large-scale database since running all HRCs requires extensive computation power. Therefore, this component consists of two algorithms for subset selection. The first one is optimized to cover different ranges of quality and bitrate. The second algorithm is optimized for HRCs that behave differently with different source contents. The two algorithms are further detailed in the accompanying DSP paper [1].

2.1.3. Quality Measure

The "Quality Measure" component consists of a collection of full reference quality measurements like Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), Multi-Scale Structural Similarity (MS-SSIM), and Visual Information Fidelity (VIFp). These measurements are integrated in this Reproducible Video Quality Analysis software package using the Video Quality Measurement Tool (VQMT) from École Polytechnique Fédérale de Lausanne (EPFL).

2.1.4. Quality Measure Analysis

The "Quality Measure Analysis" component focuses on extracting the relevant data from the full reference quality measurement database and processes it in order to perform the analysis. In this particular work we extracted frame-level MSE and PSNR values to compare the effect of temporal pooling through averaging either MSE or PSNR. Moreover, the variance of the frame-level PSNR is also computed and made available for the next visualization block. The module can be easily customized to handle different measures either present in the database or made available through files in the same format. Moreover, other indicators (e.g., moving average), can be included in the output for the visualization block.

2.1.5. Visualization

The "Visualization" block is currently a set of gnuplot command files that can be used to easily visualize the data produced in the previous step. In particular, they can automatically generate scatter plots and, with the aid of some custom-developed external modules, interpolation parameters for better visualization.

2.1.6. Verification

The creation of the large-scale database requires several steps that may cause discrepancies in the results. This ranges from different versions of system libraries when compiling the HEVC encoder to random storage media defects. In order to avoid invalid comparisons, for each bitstream of the large-scale dataset, a hash value in the form of an SHA512 checksum [2] is provided that can be verified by its GNU implementation (sha512sum).

3. Software Functionalities and Illustrative Examples

The package contains scripts that are written in different language environments but can be executed under Linux and Windows platforms. The software package content is shown in Figure 2. It contains 2 directories 'DATA' and 'SoftwareLibraries'. 'DATA' directory contains sub directories that contains the source data and the outputs of running the softwares that can be found in 'SoftwareLibraries'. First of all, the source content should be placed in (DATA/SRC). The current scripts deal with three resolutions (960x544, 1280x720, and 1920x1080). In the first module (Large-scale encoding environment), the file has to be run (SoftwareLibraries/ENC_SRC/ENC_lin.py) to generate and encode the whole coding conditions (ENC_win.py is a Windows script). The '.265' and '.txt' output files will be placed in (DATA/ENC), see part A of Figure 3. The '.265' is the bitstream file and the '.txt' is the encoding information. The quality measure module calculates the objective quality measure by running 'SoftwareLibraries/DEC.py'. The software will firstly decode the bitstream file (.265 file) and then uses the (SoftwareLibraries/VQMT_Binaries/VQMT.exe)

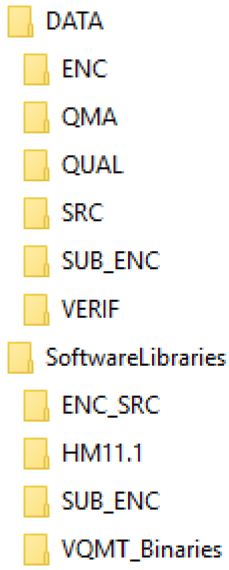

Figure 2: The content of the software packages

to calculate the PSNR, SSIM, and VIF quality measures and saves the output per frame and in average in separate files in the (DATA/QUAL) folder, as shown in the part B.1 of Figure 3. The final step in this module is to run 'SoftwareLibraries/AggregatePSNRtoCSV.py' to aggregate all the quality measures in two files, see part B.2 of Figure 3; one keeps sequence-level records, see part B.3 of Figure 3 and the second keeps the sequence and the frame levels records, see part B.4 of Figure 3.

In order to work on a subset of HRCs, the optional module "Subset Selection" has to be executed. The first step is to aggregate the input data for the MATLAB functions: 'getBitrateQualityDrivenHRCs.m' and 'getContentDrivenHRCs.m' that can be found in (SoftwareLibraries/SUB_ENC). The input data takes the form of two matrices, the first one contains the PSNR measures and the second one is the calculated bitrate. These two matrices are formatted as follows: MxN where M represents the total set of the HRCs and N represents the number of source contents. They can be aggregated from the '.txt' of the output of first module. This optional module saves the '.txt' and '.csv' files that contains the selected HRCs in the (DATA/SUB_ENC) folder, see parts C.1 and C.2 of Figure 3.

In the "Quality Measure Analysis", first sequence-level metrics are computed. The database already contains the PSNR for each frame, whereas the MSE can be computed by reversing the PSNR formula through 'script_create_other_psnr_data_and_msefile.py'. Next, 'script_create_psnr_stddev_per_frame.py' processes each HRC independently, automatically matching all the values (e.g., PSNR and MSE) related to the same HRC even if stored in different files (e.g., one per measure type) and computing the sequence-level indicators including the variance of the PSNR (σ^2_{PSNR}) which is analyzed in details in our accompanying paper [1]. Note that the software can be easily extended to include other temporal pooling strategies, or use other metrics such as SSIM. All the results are exported in a text file in comma separated value (csv) format, one line per HRC, see Figure 4.

Other utility software can perform HRC filtering operations ('script_filter_HRCs.py'), cumulative distribution function (CDF) computation while retaining all the original information ('script_compute_perpoint_cdf_function_over_PSNR_G-PSNR_A.py'), computation of the linear interpolation functions ('script_linear_interpolation.py') and of the similarity of two point cloud distributions ('script_similarity.py'), by assigning points in one graph to the nearest one in the other, then computing statistics such as average number of assigned points and average distance. In the visualize module, gnuplot command files are provided to directly generate all types of scatter plots shown in our accompanying paper [1] for immediate comparison with new research results, see Figure 4, also including the interpolating lines.

4. Impact

Since the main focus of our work is providing efficient tools for reproducibile research in the context of video quality analysis, the software has been designed and verified to exactly reproduce all the data in the database as well as all the results presented in our accompanying paper [1].

However, we would like to underline that the same piece of software, with only slight modifications, can be effectively used to pursue new research questions. For instance, for the PSNR analysis part, it is extremely easy to reuse the software with a different per-frame quality metric such as SSIM or VIFP, or testing the different results of various temporal pooling strategies. At the same time, new synthetic quality indicators can be computed and tested, e.g., by applying moving averages, mean, variances, etc. Therefore, while the software itself might appear relatively simple, in our opinion it can greatly facilitate the pursuit of new research questions that can be addressed by mining into the large set of data currently available in the database. Note that three widespread frame-level video quality metrics, i.e., PSNR, SSIM, VIFP, are already available in the database and can be immediately used for analysis, whereas other measures can be added with limited implementation effort.

The impact of this work had up to now can be seen in the following list of publications. These publications use the proposed reproducible research framework in order to run different kinds of analysis that benefit the video coding and video quality communities.

- Using the "Large Scale Encoding Environment" and the "Quality Measure" components from the provided pipeline, in [3], the authors provide a way to analyze quality measures on the sequence level to highlight the unusual behavior of these measurements. The concept of agreement between the quality measures is introduced to compare the source of disagreement for different source contents. One conclusion that can be drawn from [3] is that the VIF and the SSIM agree more often on the ordering.

- The impact of the software pipeline can also be seen in [4]. In this paper, the authors used the large scale database resulting from the "Large Scale Encoding Environment" component of this paper to compare the behavior of different quality measures in loss-impaired sequences and tried to predict the behavior of the objective measures with the help of content characteristics. The authors were thus able to reuse the provided 160 hours of video sequences without having to invest resources to encode these sequences again. For this work, the HEVC reference decoder, present in this software package, has been modified to handle packet losses. This addition allowed to investigate the performance of objective quality metrics in presence of data loss on a large scale, in particular by comparing the results of binary tests where the best sequence had to be selected.

- In [5], the author modified the last stages of the presented pipeline to predict the Peak-Signal-to-Noise Ratio full-reference measure with the help of extracted bitstream features of decoded sequences. Additionally, the authors in [5] also modified the "Subset Selection" component of the presented pipeline in order to investigate the importance of diversity of information inside a test set.

- In [6], the authors modified the "Quality Measure Analysis" component of this work in order to perform a frame-level analysis of the temporal behavior of different quality measures.

- The full software chain as proposed in this paper has been used in [1] to study the impact of different pooling strategies for the PSNR metric, i.e. the geometric and the arithmetic means.

Currently, we are not aware of any other similar research framework for video quality on such a large scale. Although the project is still at the beginning, we expect that many users will be able to take advantage of the software that we presented in this paper. Within the context of the Video Quality Experts Group (VQEG), academic partners and even some private companies have expressed interest in downloading all the software to test and analyze new metrics they are developing. The work is strongly supported by the Joint Effort Group Hybrid of VQEG that unites academic and industrial research towards improvements in video quality measures using large scale experimentation.

5. Conclusions

The presented framework has been designed in order to facilitate reproducibility of research in video quality evaluation. It is intended to provide building blocks to be expanded or reused by other researchers to test their own measures or ideas and easily compare the results in a reliable and consistent way. The framework also aims at simplifying the use of subsets of the database when testing on a large scale is computationally infeasible. To the best of our knowledge, this is the first attempt to bring exact reproducibility end-to-end in video quality evaluation research field. We hope that this framework will be further expanded through the contribution of reference implementations of new measures, indexes or simply by expanding the database by means of new content, coding coding conditions, etc. so that researchers can easily draw from that to advance their activities. All the software can be easily modified and extended, and it is released under the LGPLv3 license.

Acknowledgements

Some of the computational resources have been provided by HPC@POLITO (http://www.hpc.polito.it). Some parts of this work are supported by the Marie Skłodowska-Curie under the PROVISION (PeRceptually Optimised VIdeo CompresSION) project bearing Grant Number 608231 and Call Identifier: FP7-PEOPLE-2013-ITN. The research activities described in this paper were partially funded by Ghent University, imec, Flanders Innovation & Entrepreneurship (VLAIO), the Fund for Scientific Research Flanders (FWO-Flanders), and the European Union. Some aspects of this work were carried out using the STEVIN Supercomputer Infrastructure at Ghent University.

References

[1] A. Aldahdooh, E. Masala, G. Van Wallendael, M. Barkowsky, Framework for reproducible objective video quality research with case-study on psnr implementations, Elsevier Digital Signal Processing.

[2] P. FIPS, 180-4, Secure hash standard (SHS).

[3] G. Van Wallendael, N. Staelens, E. Masala, M. Barkowsky, Full-HD HEVC-encoded video quality assessment database, in: Ninth International Workshop on Video Processing and Quality Metrics (VPQM), 2015.

[4] A. Aldahdooh, E. Masala, O. Janssens, G. Van Wallendael, M. Barkowsky, Comparing simple video quality measures for loss-impaired video sequences on a large-scale database, in: Eighth International Conference on Quality of Multimedia Experience (QoMEX), 2016, pp. 1–6. doi:10.1109/QoMEX.2016.7498941.

[5] M. Shahid, J. Panasiuk, G. Van Wallendael, M. Barkowsky, B. Lövstöm, Predicting full-reference video quality measures using hevc bitstream-based no-reference features, in: 2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX), 2015, pp. 1–2. doi:10.1109/QoMEX.2015.7148118.

[6] A. Aldahdooh, E. Masala, G. Van Wallendael, M. Barkowsky, Comparing temporal behavior of fast objective video quality measures on a large-scale database, in: 2016 32nd Picture Coding Symposium (PCS), IEEE, 2016.

Required Metadata

Current code version

Nr.	**Code metadata description**	**Please fill in this column**
C1	Current code version	v. 1.0
C2	Permanent link to code/repository used for this code version	https://gitlab.com/gvwallen/ReproducibleQualityResearch
C3	Legal Code License	LGPLv3
C4	Code versioning system used	git
C5	Software code languages, tools, and services used	python, Matlab, shell scripts
C6	Compilation requirements, operating environments & dependencies	Windows & Linux
C7	If available Link to developer documentation/manual	https://gitlab.com/gvwallen/ReproducibleQualityResearch
C8	Support email for questions	ahmed.aldahdooh@etu.univ-nantes.fr Marcus.Barkowsky@univ-nantes.fr enrico.masala@polito.it glenn.vanwallendael@ugent.be

Table 1: Code metadata (mandatory)

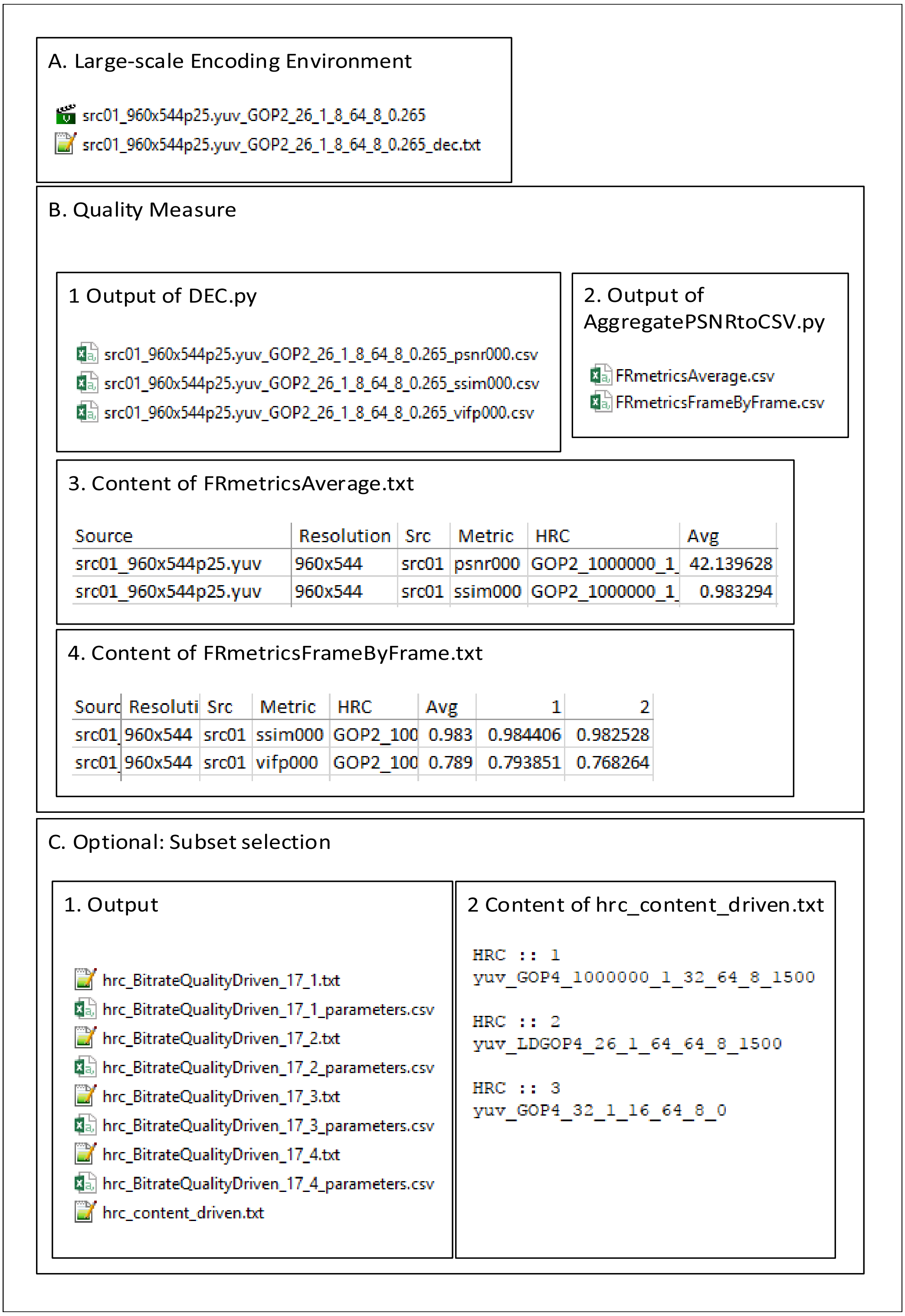

Figure 3: Illustrative example part 1. It shows the “Large Scale Encoding Environment”, “Subset Selection”, and “Quality Measure” modules.

Content of (script_create_psnr_stddev_per_frame.py) output

Source	HRC	avgpsnr PSNR_G	psnrmse PSNR_A	stddev_psnr	stddev_mse
src10_1280x720p25.yuv	GOP8_500001_1_32_64_8	35.140386	34.598643	2.144403	12.832856
src10_1920x1080p25.yuv	GOP8_16000000_1_64_64	47.464699	47.228376	1.412654	0.445434

Output graph of (graph_scatter__PSNRvar_vs_PSNR_interpolated.gplot)

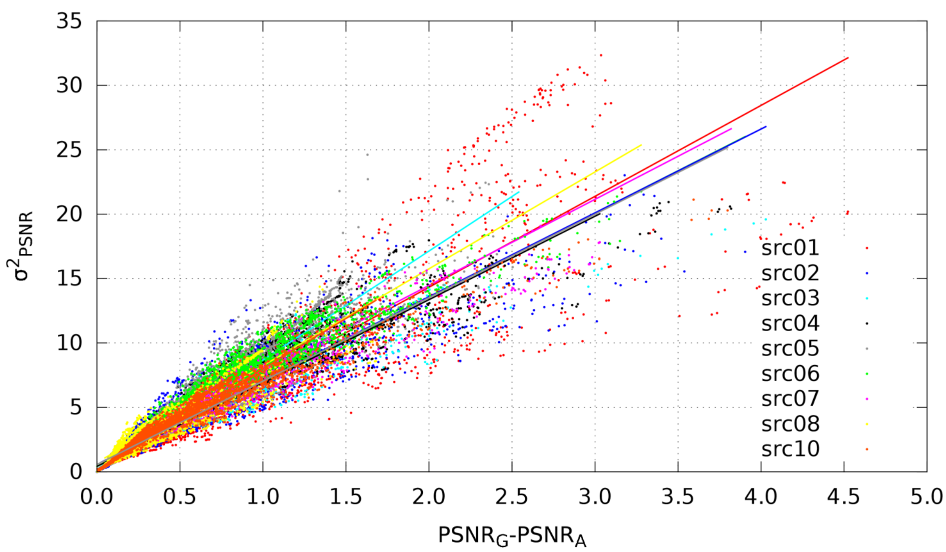

Figure 4: Illustrative example part 2. It shows the Quality measure analysis, and the visualize modules.