

An Automated Testing Environment For The ITU-I/ISO-IEC Reference Video Encoders

José Raimundo Barbosa*, Jean Felipe Fonseca de Oliveira[‡] Carlos Danilo Miranda Regis*, Ruan Delgado Gomes[†]

*Instituto Federal da Paraíba - Campus João Pessoa

[†]Instituto Federal da Paraíba - Campus Campina Grande

[‡]Sansung, São Paulo, Brasil

E-mail: jraimundo@lavid.ufpb.br, jeanfelipefonseca@gmail.com, danilo.regis@ifpb.edu.br, ruan.gomes@ifpb.edu.br

Abstract—The ITU-T/ISO-IEC H.265/HEVC is the state-of-the-art video codec, focusing in Ultra-High Definition video (UHD) and it is 50% more efficient in bit-rate compression than its predecessor, the H.264/AVC. However, this efficiency requires a high computational cost, which may compromise the use of the encoder in limited machines or in real time applications that use UHD videos. Several studies look for ways to enhance the HEVC encoder and reduce the computational cost. However, to develop optimized versions of the encoder it is necessary to perform tests that can last for hours and usually requires manual collection of coding information. In this work, a tool to perform automated tests of HEVC encoders was developed. The tool was implemented in a parallel architecture that allows the execution of several instances of the encoder simultaneously. It collects the informations from the encoders and executes objective metrics to evaluate its efficiency, organizes the results and generates reports. This work intends to provide an useful solution to help video coding researchers analyzing encoders in long sequences of tests.

I. INTRODUCTION

Currently, about three-quarters of annual data traffic on the Internet is assigned to video content, according to Cisco's Visual Networking Index report. In 2020 this value will be about 80% [1]. One of the contributing factors is the increasing in video quality, which enables new experiences on Internet-broadcast television channels, video conferencing and on-demand video transmission services such as YouTube, Amazon and Netflix, which already distribute videos in UHD [2].

The amount of pixels in UHD videos leads to an significant improvement in the perception of image details, allowing new experiences for viewers, but it also represents challenges to current market [2]. For example, a UHD 4K video has a resolution of 4096x2160 pixels, while previous standards such as Full HD have 1920x1080. Considering a raw uncompressed UHD 4K video sequence, with 24 bits per pixel and 24 frames per second, it would require a bandwidth of 5 Gbit/s [3]. These settings represent challenges in the storage and transmission of such videos.

In order to handle the intense and growing traffic of UHD videos over the Internet and to address the bandwidth limitations, it was necessary to develop codecs capable of the challenging task of significantly reduce transmitted data volume without sacrifice the perceived visual quality [4][5] [6]. The ITU-T H.265 [7], also known as High Efficiency Video

Coding (HEVC), is the state-of-the-art codec which focus primarily on the newest UHD video technologies and presents a compression efficiency 50% higher than its predecessor, H.264/Advanced Video Coding (AVC) [8][9][10]. To reach the aforementioned performance, the HEVC standard implements several enhancements in relation to the H.264, such as the Quadtree Partitioning Scheme and the increased number of intra prediction modes (HEVC has 33 modes while H.264 has 9). However, all these changes leads to a computational cost up to four times higher than H.264. This scenario presents as a challenge for situations that demand real-time video reproduction or in devices with limited processing power [11][12].

To promote the HEVC optimization research, the JCT-VC committee provides the HM reference codec source code which features a full implementation of HEVC specifications that enables to carry out studies on the efficiency and performance of the codec and to develop new features and optimizations [13]. However, the performance evaluation of codecs developed based on the HM reference source code is a time-consuming process, since each test implies in the execution of several instances of the evaluated encoder and the reference encoder, which can last for several hours. After execution, is necessary to compare the UHD videos using objective or subjective metrics. With the comparisons it is possible to assess the efficiency and performance of the proposed new encoders [6].

This paper proposes and describes an automated testing tool of HM based encoders. The implementation of this tool features a parallel architecture in which multiple instances of the encoder can be simultaneously executed. To calculate the encoder efficiency and the performance difference between the encoders, the Bjontegaard metric is used. This metric determines the difference of the efficiency of the codecs from the calculation of the mean difference of two curves adjusted by four QP (quantization parameters) points [14]. The PSNR, SSIM and PW-SSIM[26] metrics are also used. For the formulation of the standardized test cases, the recommendations of JCT-VC L.1100 were followed [15]. With the tool, the comparison between the [4] codec and reference codec [13] was obtained and reduction of the test time by 86% in a simple test case was verified, in which one video and one configuration were used, totalizing the execution of 8 instances.

II. STANDARDIZATION OF TESTS

The Joint Collaborative Team on Video Coding (JCT-VC) provided the open source HM reference encoder, which offers a implementation of HEVC in C++ [15]. The HM code provides an environment for the development of new features or optimizations, such as those developed in [4], [11], [12] and [16]. In conjunction with the encoder, JCT-VC also provides the recommended settings and videos for standardized test suites. These settings are presented later.

After the encoding, the HM generates, in addition to the encoded video, a record file containing important information about the encoding process such as bit rate, total encoding time, output video file size and PSNR for each YUV color matrix (Y = Luma, U = Chroma Blue and V = Chroma Red).

The way the tool was developed also enables to perform tests with the Joint Exploration Model (JEM) reference encoder [17]. This encoder was developed based on HM, and was presented in [18] as the future of the environment for standardized development of HEVC video compression technology, including its current extensions and near-term extensions for screen content coding and high-dynamic-range coding.

To ensure common testing conditions, the JCT-VC committee has developed the JCTVC-L1100 documentation [19], which describes the settings and contents for standardized tests that explore different situations. This standardization is applied in the development of new features or encoder optimizations developed using HM. The settings discussed in the document are:

- All Intra: All frames are encoded independently with only intra-frame prediction.
- Random Access (RA): Coding with high rates of compression. The coding order and the output order of the frames differ, inducing a coding delay. This setting represents broadcast and streaming applications.
- Low Delay P (LP): Coding without the structural coding delay presented when using RA parameters. This configuration is evaluated using only P-frames (uni-directional prediction).
- Low Delay B (LB): Similar to LP scenarios, but this configuration is evaluated using only B-frames, (unidirectional and bidirectional prediction).

These settings specify situations for videos with 8 bits per color component, and with 10 bits per color component. In the documentation, the videos present in *ftp://hevc@ftp.tnt.uni-hannover.de/testsequences/* are also recommended, requiring a registration to access the videos. Finally, the quantization values are specified as: 22, 27, 32 and 37. In the proposed tool, these values are already configured with these quantization values, but it is possible to add other quantization values.

III. OBJECTIVE METRICS

The evaluation of the quality of the videos can be carried out subjectively, that is, from the human evaluation [20]. However, this type of evaluation may be ineffective in detecting small image debris, which is generally difficult to perceive in the human eye, and may be exhaustive in long applications or in the case of real-time evaluation [21][22]. However, objective

video evaluation techniques work better because they are faster and cheaper and indicate the existence of slight degradations [22].

In this section, the objective metrics used in this work are discussed.

A. PSNR Metric

The Peak Signal to Noise Ratio (PSNR) is one of the most used metrics because of the ease of calculation and the low computational cost [23]. This method expresses the maximum ratio of a signal and the noise of this signal and is constantly used to measure the structural quality of images. For this it is necessary to formulate the mean square error for the digital representation of monochromatic images I and K of size m x n, the MSE is defined by the Equation 1.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2 \quad (1)$$

In this way, the PSNR is represented by 2, where MAX indicates the maximum possible pixel value in an image.

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) = 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \quad (2)$$

The Equation 2 results in a value between 0 and ∞ . In the literature a range between 0 and 50 is usually estimated, with 0 being the worst case and 50 being the best.

B. SSIM Metric

The Structural Similarity Index (SSIM) is a consolidated and constantly used model for video quality comparison. It is based on the assumption that the human visual system is highly adapted to extract structural information from images [24]. Therefore, each pixel has a strong dependence on the others and this dependence increases with the proximity. This dependence presents important information about the structure of the objects in the image and that quantifying the structural change of an image can provide a good approximation to the perceived quality [25].

To identify the structural information, the SSIM metric uses a statistical approach based on the average luminance and $n \cdot n$ block contrast of the image. The mean (μ), standard deviation (σ^2) and covariance (σ_{fg}) are calculated for each bloc, the mean and standard deviation are approximate estimates of the luminance and contrast of the image, respectively. Covariance is the measure of how much a signal is different from the other.

SSIM uses the measure of structural distortion rather than the error itself. If $x = \{x_i | i = 1, 2, \dots, N\}$ represents the original signal and $y = \{y_i | i = 1, 2, \dots, N\}$ represents the distorted signal, where i is the pixel index value. The structural similarity index can be calculated according to Equation 3 [4][25].

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (3)$$

In the Equation 3, μ_x represents the average of x , μ_y represents the average of y . The variance of x is represented by σ_x^2 and the variance of y by σ_y^2 . The $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ are the variables to stabilize the division, $L = 2^n - 1$ is the dynamic range of bits and $k_1 = 0.01$ and $k_2 = 0.03$ by default.

C. PW-SSIM Metric

The Perceptual Weighting Structural Similarity Index (PW-SSIM) uses a weighting-based approach to evaluate image quality, giving more importance to visually more important regions [26]. For this the magnitude of the gradient vectors of the original video is calculated using Sobel masks [27], then a frame is generated in which the pixel values are the magnitudes of the gradients. Then, this frame is partitioned into blocks of 8×8 pixels and for each block the Spatial Perceptual Information (SI) is calculated [28]. The SI is expressed by Equation 4.

$$SI = \left(\frac{1}{N-1} \sum_{i=1}^N (\mu_S - S)^2 \right)^{\frac{1}{2}} \quad (4)$$

where N is the number of pixels per blocks and μ_S represents the average magnitude of the gradient in a block. Finally, the PWSSIM is calculated using.

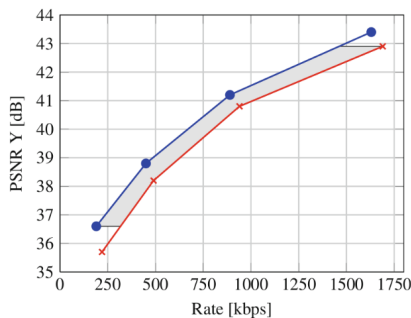
$$PWSSIM(f, h) = \frac{\sum_{d=1}^D SSIM_d(f, h) \cdot SI_d}{\sum_{d=1}^D SI_d} \quad (5)$$

where D is the number of blocks, and f is the representation of a 2D video and h represents a 2D video degraded.

D. Bjontegaard Metric

The Bjontegaard metric is considered the state-of-the-art with regards to the evaluation of video coding mechanisms and is one of the most widespread metrics in the current literature [14][29]. It allows to obtain the values for overall bit rate (BD-Rate) in percent for two different encoding algorithms considering the same PSNR or overall video quality difference (BD-PSNR) in decibels, between two encoders, considering the same bit rate.

Figure 1: Interpolation of the BD-Rate [29].



In [29], the BD-Rate is exemplified by means of Figure 1, where the curves are further drawn by linear interpolation. For the BD calculations, the rate axis is logarithmized and the rate-distortion curves are interpolated by an advanced interpolation method. The overlapping portions of the rate-distortion curves are used for calculation of the BD values. Depending on the range of overlap of the two curves, the resulting BD measurements get more or less meaningful. Therefore, a careful interpretation of the resulting numbers is required.

IV. RELATED WORKS

In the literature it is possible to find several studies related to HEVC optimization techniques. Usually the methodological characteristics used in the development of the new technique as metrics and videos used are presented, besides the obtained results. However the test stage is vaguely mentioned or simply has its description ignored. As it happens in [4], an alternative is presented that uses machine learning to predict the partitioning of the macroblocks and to avoid the extra processing, and the proposed solution is compared with the HM reference encoder. In [30], a parallel approach is used to accelerate coding by distributing multi-core processor efforts. And in [5], it is proposed an optimized partitioning algorithm, which refer to a scenario where the coded videos are processed by metrics and then the results are compared, without the aid of a tool for extraction and calculation.

The alternative proposed in [31] offers an encoder comparison service, in which several configurations are predefined for the tests. For this, it is necessary to submit the encoder to one of the paid or free services offered. Considering the best service, the premium, includes the SSIM, PSNR objective metrics and subjective metrics psycho visual enhancements, also includes the configurations High Speed, Normal and High. Quality, and presets analysis of sythetic motion, distortion in tail area and spatially variable noise. However, this alternative requires payment for more complex tests and also limits testing to a fixed number of configurations and test cases. In addition, more current and advanced metrics like BD-Rate and BD-PSNR are not mentioned. The postponed solution also uses as reference encoders implemented based on X.265, for UHD video tests, this encoder is constantly mentioned in the literature, however the test scenarios proposed by the JVET specify HM encoders. These specifications help to standardize and document studies on the enhancement of the HEVC encoder. The free service offers tests similar to the paid one, but with the reduced number of videos, metrics and reports.

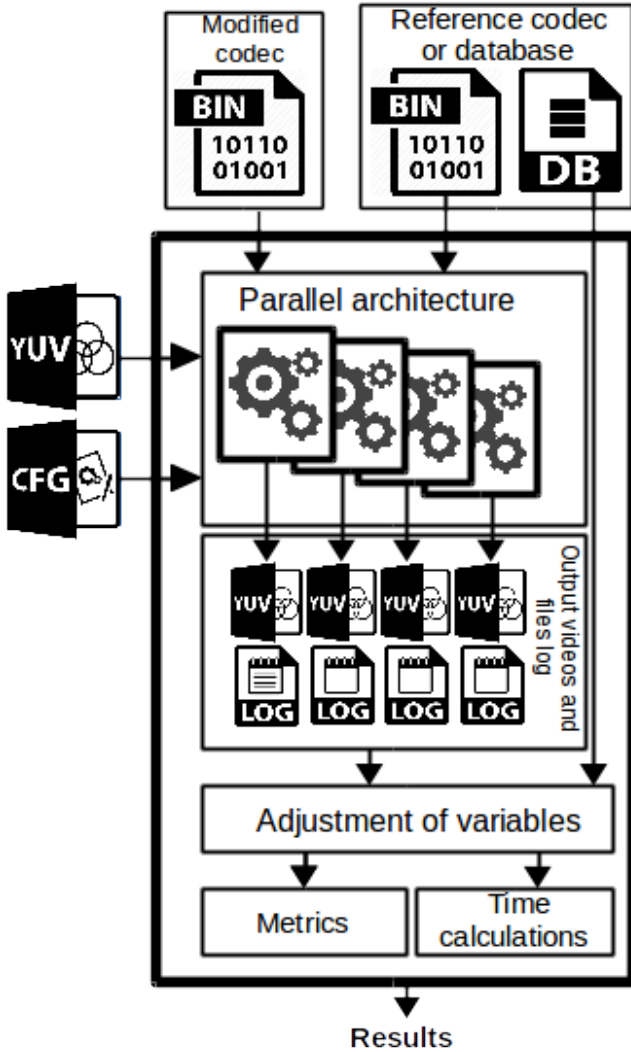
V. TOOL DESCRIPTION

To verify the efficiency of an encoders it is necessary to perform several tests, in which the encoder under test is executed in several coding scenarios, and then the results for each scenario are compared with the results obtained using the reference encoder. Each test sequence of an encoder can last for hours, depending on the videos and test settings, which hinders the development stage. The comparison is made based on values such as bit rate, PSNR, coding time, and so on.

These values are generated automatically by the HM encoder and are saved in log files. However, for their use it is necessary to manually collect in each log file, which is a repetitive work and can result in errors from human failure.

The tool developed in this work executes instances of the modified encoder and reference encoder in all required configurations and then automatically collects all the information necessary for the comparison. With the data collected from the log files the calculation of the Bjontegaard [32] metric is performed. HM-based encoders offer a decoded output video, these videos are used by objective metrics SSIM, PW-SSIM, and so on. to calculate the image quality after encoding. The tool architecture is shown in Figure 2.

Figure 2: Tool architecture.



The modular architecture of the tool enables other metrics to make use of the encoding values. To reduce the time of the tests, the tool was implemented in a parallel architecture in which multiple instances of the encoder are executed

simultaneously, thus considerably reducing the time of a test. For this, is necessary a adequate management is required so that the execution of different threads does not compromise the integrity of the values used to calculate the metrics, time and other information present in the sequence of tests.

To prevent data loss during an unexpectedly stopped execution, a tool log file is generated that saves the current state of the test and makes it possible to return from where it left off. It is also possible to save the data of the tests performed by the encoders, thus avoiding to repeat the process and thus to promote the construction of a database with the results of the tests.

To execute the tests, the tool receives the parameters specified in Table I, these parameters indicate the settings that will be addressed equally by the modified encoder and the reference encoder that will generate different outputs. If the parameters are not informed, default values will be used that will execute the tool using a CIF video (352x288) and a one of the standard settings (Random Access Encoding). These parameters have minimum specifications only for checking the operation of the tool. It is important to check the memory and thread limitations before running the tool, it is recommended to run one thread for every two GBs of RAM. In case of an unexpected interruption, the execution of the tool with the -bcp flag activates the backup and the test will return to the moment of interruption.

Table I: Parameters for tool execution.

Command	Example	Description
./CodeCTest	–	Tool executable
-thr	2	Number of threads
-vin	1 path/video.yuv	Number of video followed by the video files
-cfg	1 path/File.cfg	Number of configuration followed by the configurations files
-outl	path/out/logs/	Path where log files will go
-outv	path/outvideos/	Path where encoded videos will go
-eva	path/BinEncoder	Rated encoder binary
-ref	path/BinEncoder	Reference encoder binary
-fr	30	Frame rate
-f	130	Number of frames
-q	4 22 27 32 37	Number of quantization followed by the quantization values
-wdt	3840	Video width
-hgt	2160	Video height
-bcp	–	Backup flag

To calculate metrics such as PSNR, SSIM among others, it

is necessary to execute at least one coding with the modified coder and one of the reference coder, using the same configuration and the same input video, and finally calculate the metrics on the resulting videos. In the case of the metric proposed by Bjontegaard, four executions are required for each encoder, one execution for each QP, totaling eight executions. A execution of a simple test case, in which there are eight executions, for each pair of encoder videos with the corresponding QP, the conventional metrics are calculated, and at the end of the eight executions the Bjontegaard metric is calculated. The tool also enables more extensive test case where you can increase the number of QP, number of videos or settings. The output file structure of the testing tool presents the key information about the execution of each instance, in addition to the total time values and the values of the metrics. A log file example is shown in Figure 3.

Figure 3: File log example.

```

ENCODER TEST LOG:
CONFIG .....: test/cfg/config.cfg
VIDEO .....: ../Library.yuv
Chroma format .....: 4:2:0
Color depth .....: 8
Width .....: 3840
Hight .....: 2160
Frames .....: 300
Frame rate .....: 130
-----
QP: 22      Evaluate      Reference
PSNR .....: 11.7142      11.7142
SSIM .....: 0.579638     0.579638
PW-SSIM ...: 0.581134     0.581114
Time .....: 13890.753    28705.256

QP: 27      Evaluate      Reference
PSNR .....: 11.6148      11.6148
SSIM .....: 0.568489     0.568489
PW-SSIM ...: 0.570788     0.570748
Time .....: 12080.048    26321.021

QP: 32      Evaluate      Reference
PSNR .....: 11.5353      11.5353
SSIM .....: 0.558388     0.558388
PW-SSIM ...: 0.560307     0.560271
Time .....: 11594.021    25618.921

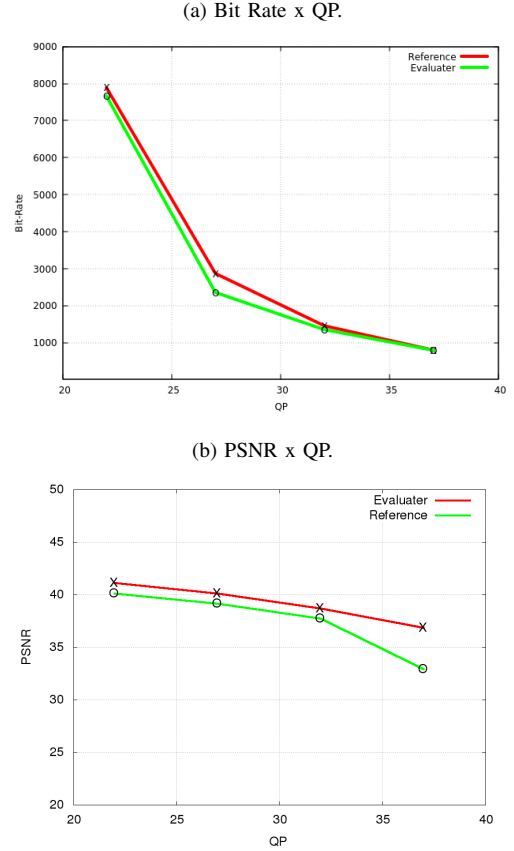
QP: 37      Evaluate      Reference
PSNR .....: 11.3883      11.3883
SSIM .....: 0.547443     0.547443
PW-SSIM ...: 0.549217     0.549226
Time .....: 11461.959    25333.508
-----
Bjontegaard AVG .....: 0.0165278
Bjontegaard Rate .....: -0.87248
Total time: 0 : 8 : 1 : 54

```

This type of visualization helps developers to identify the main differences in the final result between their respective encoders and the original version.

It is also possible to observe the values of Bit Rate per QP used in the test in the Figure 4a, and also the PSNR values by QP in the Figure 4b, these graphs are generated automatically after the tool execution.

Figure 4: Chart examples.



VI. RESULT

To validate the tool, the modified version of the HM.16 encoder proposed in [4] was evaluated, using the original version of the encoder as reference. The test was performed on a computer with AMD FX-8320 processor (8 cores), 32 GBs of RAM and Ubuntu 16.04 operating system. The 4K sequences used are available in [33], this videos has a frame rate of 30 fps, and 8 bits per color component, in a total of 600 frames.

In a simple test case, the total time spent executing the encoders and calculating the metrics using multi-threaded processing was 8 hours while the same test performed in a conventional scenario, ie running one instance of the encoder at a time, running the metrics independently and collecting data from the log files, was 1 day, 19 hours and 28 minutes, without considering the time for data collection in the log files.

VII. CONCLUSION

For more complex tests, the implementation of a distributed version of the tool is in progress, which will allow the execution of more instances of the encoder according to the configuration and quantity available machines. Online services are also being developed, in which researchers and developers can submit the executable program of their encoders and select the configurations so that the tests are performed on a robust server. With this proposal also intends to build a database fed with the data of the tests of the users, this database will contribute to other researchers.

REFERENCES

- [1] CISCO. “Cisco VNI Forecast and Methodology, 2015-2020”. Technical report, CISCO, 2016. Disponible: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html>, Access: 8 January 2016.
- [2] M. Cheon and J. S. Lee. “Objective Quality Comparison of 4K UHD and Up-Scaled 4K UHD Videos”. In *2014 IEEE International Symposium on Multimedia*, pp. 78–81, Dec 2014.
- [3] R. D. Gomes, Y. G. G. d. Costa, L. L. Aquino Júnior, M. G. d. Silva Neto, A. N. Duarte and G. L. d. Souza Filho. “A Solution for Transmitting and Displaying UHD 3D Raw Videos Using Lossless Compression”. In *Proceedings of the 19th Brazilian Symposium on Multimedia and the Web*, WebMedia '13, pp. 173–176, New York, NY, USA, 2013. ACM.
- [4] J. F. F. de Oliveira. “Técnica de Aprendizagem Automática Aplicada a um Codificador HEVC em Tempo Real”. In *Tese (Doutorado) - Curso de Pós Graduação em Engenharia Elétrica*, p. 112, 2016. Departamento de Engenharia Elétrica, Campina Grande.
- [5] C. C. Wang, Y. C. Liao, J. W. Wang and C. W. Tung. “An Effective TU Size Decision Method for Fast HEVC Encoders”. In *2014 International Symposium on Computer, Consumer and Control*, pp. 1195–1198, June 2014.
- [6] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy and M. Manohara. “Toward A Practical Perceptual Video Quality Metric”. Technical report, Netflix, 2016. Disponible: <http://techblog.netflix.com/2016/06/toward-practical-perceptual-video.html>, Access: 10 January 2016.
- [7] I. T. Union. “Recommendation H.265: High Efficiency Video Coding”. Recommendation, ITU, 2016. Disponible: <https://www.itu.int/rec/T-REC-H.265>, Access: 18 January 2016.
- [8] F. Bossen, B. Bross, K. Suhring and D. Flynn. “HEVC Complexity and Implementation Analysis”. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1685–1696, Dec 2012.
- [9] P. Hanhart, M. Rerabek, F. D. Simone and T. Ebrahimi. “Subjective quality evaluation of the upcoming HEVC video compression standard”. In *Applications of Digital Image Processing XXXV*, 2012.
- [10] G. J. Sullivan, J. R. Ohm, W. J. Han and T. Wiegand. “Overview of the High Efficiency Video Coding (HEVC) Standard”. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.
- [11] J. S. Yoon, C. Lee, C. Park, G. Lee, K. Lee, S. Roh, M. Jeon, Y. Jung, J. Oh and J. A. Lee. “An H.265/HEVC codec for UHD(3840 x 2160) capturing and playback”. In *2013 International SoC Design Conference (ISOCC)*, pp. 218–220, Nov 2013.
- [12] G. Correa, P. Assuncao, L. Agostini and L. A. da Silva Cruz. “Performance and Computational Complexity Assessment of High-Efficiency Video Encoders”. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1899–1909, Dec 2012.
- [13] I. T. Union. “HEVC Test Model (HM)”. Recommendation, ITU, 2016. Disponible: <https://hevc.hhi.fraunhofer.de/HM-doc/>, Access: 18 January 2016.
- [14] G. Bjøntegaard. “Document VCEG-M33: Calculation of average PSNR differences between RD-curves”. In *ITU-T VCEG Meeting*, 2001.
- [15] F. Bossen, D. Flynn, K. Sharman and K. Suhring. “HM Software Manual”. *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*, p. 30, Oct 2015.
- [16] R. Weerakkody, M. Mrak, V. Baroncini, J. R. Ohm, T. K. Tan and G. J. Sullivan. “Verification testing of HEVC compression performance for UHD video”. In *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1083–1087, Dec 2014.
- [17] F. Bossen, D. Flynn, K. Sharman, K. Suhring and X. Li. “JVET JEM software”. Technical report, JVET, 2015. Disponible: <https://jvet.hhi.fraunhofer.de/>, Access: 8 April 2017.
- [18] JVET. “Algorithm Description of Joint Exploration Test Model 1 (JEM 1)”. Technical report, JVET, 2015. Disponible: <http://mpeg.chiariglione.org/standards/exploration/future-video-coding/n15790-algorithm-description-joint-exploration-test-model>, Access: 8 April 2017.
- [19] F. Bossen. “Common HM test conditions and software reference configurations”. Technical report, JCT-VC, 2013. Disponible: <http://phenix.it-sudparis.eu/>, Access: 18 January 2017.
- [20] ITU. “INTERNATIONAL TELECOMMUNICATION UNION: ITU-T recommendation P.910, subjective video quality assessment methods for multimedia applications”. Recommendation, Sep 1999.
- [21] C. R. D. Estrada. “Avaliação automática de qualidade de video-conferências de alta definição”. Technical report, 2009.
- [22] J. V. M. Cardoso, A. C. S. Mariano, C. D. M. Regis and M. S. Alencar. “Comparação das Métricas Objetivas de Qualidade de Vídeos Baseadas na Similaridade Estrutural e na Sensibilidade ao Erro”. *Revista de Tecnologia da Informação e Comunicação*, vol. 1, no. 2, pp. 33–40, Apr 2012.
- [23] C. D. de Miranda Regis, I. de Pontes Oliveira, J. V. de Miranda Cardoso and M. S. de Alencar. “Design of Objective Video Quality Metrics Using Spatial and Temporal Informations”. *IEEE Latin America Transactions*, vol. 13, no. 3, pp. 790–795, March 2015.
- [24] J. V. de Miranda Cardoso, A. C. S. Mariano, C. D. M. Regis and M. S. Alencar. “Comparação das Métricas Objetivas de Qualidade de Vídeos Baseadas na Similaridade Estrutural e na Sensibilidade ao Erro”. *Revista de Tecnologia da Informação e Comunicação*, 2012.
- [25] Z. Wang, L. Lu and A. C. Bovik. “Video quality assessment using structural distortion measurement”. In *Proceedings. International Conference on Image Processing*, volume 3, pp. III–65–III–68 vol.3, 2002.
- [26] C. D. de Miranda Regis, I. de Pontes Oliveira, J. V. de Miranda Cardoso and M. S. de Alencar. “Design of Objective Video Quality Metrics Using Spatial and Temporal Informations”. *IEEE Latin America Transactions*, vol. 13, no. 3, pp. 790–795, March 2015.
- [27] A. Furnari, G. M. Farinella, A. R. Bruna and S. Battiatto. “Generalized Sobel Filters for gradient estimation of distorted images”. In *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 3250–3254, Sept 2015.
- [28] J. F. F. de Oliveira, C. D. M. Regis and M. S. de Alencar. “Performance Evaluation of the PWSSIM Metric for HEVC and H.264”. *International Conference on Communication, Management and Information Technology*, vol. 65, no. 9, pp. 115–124, 2015.
- [29] M. Wien. “High Efficiency Video Coding: Coding Tools and Specification”. *Signals and Communication Technology*, 2015.
- [30] Y. Wang, X. Guo, Y. Lu, X. Fan and D. Zhao. “GPU-based optimization for sample adaptive offset in HEVC”. In *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 829–833, Sept 2016.
- [31] M. G. M. Lab. “Video Codec Analysis and Testing”. Technical report, MSU Graphics Media Lab, 2012. Disponible: http://compression.ru/video/codec_comparison/codec_analysis.html, Access: 10 August 2017.
- [32] G. Bjøntegaard. “Calculation of Average PSNR Differences Between RD-curves”. *ITU - Telecommunications Standardization Sector*, 2001.

- [33] SJTU. “SJTU 4K Video Sequences”. Technical report, SJTU, 2013. Disponible: <http://medialab.sjtu.edu.cn/web4k/index.html>, Access: 8 April 2017.