Revisiting Bjontegaard Delta Bitrate (BD-BR) Computation for Codec Compression Efficiency Comparison

Nabajeet Barman
Brightcove
London, UK
nbarman@brightcove.com

Maria G Martini
Kingston University
London, UK
m.martini@kingston.ac.uk

Yuriy Reznik
Brightcove
Boston, USA
yreznik@brightcove.com

ABSTRACT

Bjontegaard Delta Bitrate (BD-BR), proposed in 2001, remains one of the most widely and also misunderstood tools for the computation and comparison of codec compression efficiency of two or more video codecs. Initially proposed for BD-Rate and BD-PSNR savings calculation using PSNR as the choice of objective quality metric, many works in recent years have calculated and reported similar measurements using other objective metrics such as SSIM, VMAF and even MOS. Their understanding and usage, however, remains limited to mostly standardization activities with very limited work in the scientific literature studying the performance of such metrics under different conditions. Towards this end, in this paper we present three different studies related to BD-BR computation, both in terms of bitrate and quality savings. Different open source implementations, extensions and alternatives are evaluated on two different datasets, considering three objective quality metrics (PSNR, SSIM and VMAF) and subjective quality ratings (MOS scores). Based on various results and observations from this work, we present a set of recommendations on the use of existing BD-BR metrics as well as present various insights and opportunities for collaborative work on the development of more effective tools for codec compression efficiency evaluation and comparison. Additionally, all the evaluated metrics, their implementations and sample datasets used in this work are provided as an open source dataset[1] for reproducibility of the results and future work in this direction.

CCS CONCEPTS

• **Information systems → Multimedia streaming**.

KEYWORDS

Video Compression, Codec Comparison, BD-Rate, BD-Quality, BD-PSNR, Rate–distortion Curves, Bjøntegaard delta

ACM Reference Format:
Nabajeet Barman, Maria G Martini, and Yuriy Reznik. 2022. Revisiting Bjontegaard Delta Bitrate (BD-BR) Computation for Codec Compression Efficiency Comparison. In *Mile High Video Conference (MHV '22), March 1–3, 2022, Denver, CO, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3510450.3517289

[1]https://github.com/NabajeetBarman/BD-BR

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MHV '22, March 1–3, 2022, Denver, CO, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9222-8/22/03…$15.00
https://doi.org/10.1145/3510450.3517289

1 INTRODUCTION

Recent years have seen a rise in OTT streaming services such as Netflix, YouTube, and Twitch. The increasing popularity of such services can be attributed primarily to increased network bandwidth and increased compression efficiency supported by the proliferation of video playback devices such as smartphones, tablets, and TVs, leading to the user expectation of any time, anywhere, any device streaming. Video compression is one of the most efficient ways to reduce a media file size for faster transmission and delivery over the network. In the past almost 20 years, many advancements have been made towards the development of more efficient codecs, from H.264/AVC in 2003 to the more recent H.266/VVC [13]. However, with an increasing number of codecs being proposed, it is often tricky for the industry to decide on if and which codec to choose/adopt. For the adoption of a new codec, many different factors are required to be considered, including but not limited to compression efficiency, speed, cost, application, number of encodes required, etc. One of the most commonly used measurements for comparison of codec compression efficiency is using bitrate and quality savings as calculated using the BD-BR functions first proposed in 2001 [4]. Over the past 20 years, many improvements and open source implementations of the Bjontegaard Delta Bitrate (BD-BR) metrics have been proposed. Even though BD-BR metrics have been widely accepted and used for rate or quality savings computation, proper documentation, and a systematic evaluation of the BD-BR metric and its variations, given the era of newer codecs, quality metrics, as well as datasets is still missing. Also, many different works have indicated that the results shown by the BD-BR metric are often quite different in real-world measurements and/or results obtained from subjective experiments. Correctness of the results is critical for the fairness of the compared video codecs. As an effort in this direction, in July 2020, an ITU-T Technical paper was published which describes BD-BR computation for video coding experiments [11]. As mentioned in the report, the scope of the work was limited only to a conceptual level overview of the metrics, reasons behind some of the choices, references to technical papers and discussion of some situations where the results should be interpreted cautiously. However, no evaluation was performed and no "recommendations" as such were provided.

Towards this end, in this paper, we present three different studies related to BD-BR computation, both in terms of bitrate and quality savings, which are:

(1) Study I: In the first study, we evaluate the performance of the implementations used in standardization activities by JVET

and ITU-T [5, 7] and various open-source BD-BR metric implementations for a comparative evaluation of the results obtained on a dataset considering the RD curves of two videos encoded using two codecs. Two objective quality metrics (PSNR and VMAF), different ranges of bitrate points are also considered.

(2) Study II: In the second study, using an open-source dataset, the three different BD-BR metric implementations (selected using results of Study I) are evaluated using data from an open-source dataset consisting of RD curves of four different videos encoded using the two state-of-the-art codecs. Three objective quality metrics (PSNR, SSIM and VMAF) are considered for BD-BR calculations.

(3) Study III: Lastly, we compare the performance of the three BD-BR metrics and SCENIC (an open-source alternative to BD-BR metric when considering subjective ratings) for codec comparison efficiency computation using subjective (mean opinion score, MOS) scores from the same dataset.

1.1 Background on Evolution of the BD-BR metric

1.1.1 Orignal BD-BR Metric. In 2001, Gisle Bjøntegaard proposed a method to calculate the average PSNR difference between two RD curves [4]. The basic proposal was to fit a curve through 4 data points and then find an expression for the integral of the curve. The average bitrate "savings", referred to as BD-Rate, was then calculated as the difference between the integrals divided by the integration interval. Since the higher bitrates in a "normal" RD plot dominated the bitrate savings, it was proposed to take the logarithm of the bitrates, resulting in *dB* units on both axes. This also allowed for the "reciprocity" of calculation of change in bitrate or change in PSNR, thus allowing for calculation of both Quality (PSNR) savings and Bitrate savings. The quality savings is referred to as BD-Quality. Henceforth, this function will be referred to as BD-BR_Original.

1.1.2 Modified BD-BR Metric. The original BD-BR function used a third-order polynomial interpolation for the curve fitting with logarithmic bitrate scale. Third-order polynomial is a cubic fitting function, but there are only four adjustable parameters, and the objective is to try to get the best fit (but not necessarily a perfect fit) to a fixed number of (x, y) data points (four in the proposed metric). The values of those fitting parameters are calculated while trying to optimise some "goodness of fit" criterion, such as the total squared error. However, some studies later found that the third-order curve fitting method is not always stable and hence, for more reliable results, the use of the piecewise cubic interpolation method along with logarithmic bitrate scale was instead proposed in 2009 [6] and its implementation was provided in [1][16]. In the piecewise cubic case, one finds an exact fitting cubic which has to go through the consecutive data points, such that all the "pieces of cubics" all join up - i.e. the result is continuous at the joining points, hence resulting in more stable results. This modified BD-BR function with piecewise cubic interpolation will be referred to as BD-BR_Piecewise in the rest of this paper.

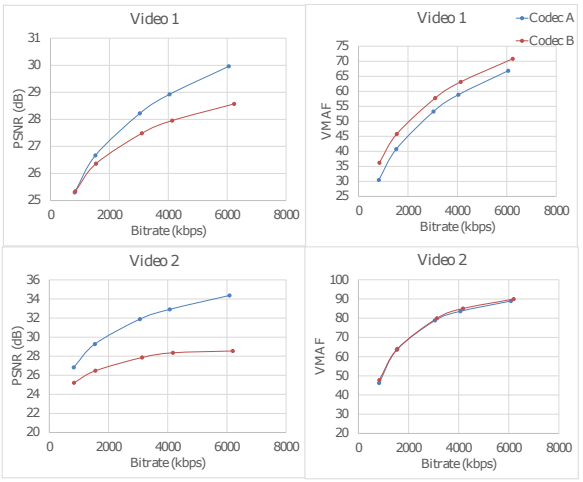

Figure 1: Quality vs Bitrate curves for the two videos from Dataset 01.

1.1.3 Extended BD-BR Metric. More recently, in 2017 Alexis et al. in [16] proposed a new excel template for the calculation of BD-BR values for more than 4 data points and proposed various modes to the original function which allows for the calculation of values by extrapolation (or interpolation) when the RD curves are not originally overlapping. The contribution includes an excel template allowing for the computation of the old BD-BR formulation along with the proposed formulation. This function will be referred to as BD-BR_Extended in the rest of this paper. Unless mentioned otherwise, the default mode (None) is used in this work.

2 STUDY 1: COMPARISON OF DIFFERENT IMPLEMENTATIONS

As discussed earlier, in the original BD-BR metric, a 3rd order polynomial interpolation was used for curve fitting. However, later it was realized that the polynomial interpolation might result in unreliable results. Instead, the use of the piecewise cubic interpolation method was proposed later for more stable and reliable results. Most of the "official" contributions have been limited to Excel macros and over time many other implementations are being made available in other programming languages. Also, the original implementations were designed to consider only four data points [16]. However, many works have reported BD-BR results considering RD curves with more than 4 data points. Hence, in our first study, we evaluate some of the publicly available open-source implementations for BD-BR calculation and try to understand their capabilities in terms of handling more than 4 DPs and the BD-BR function they use.

2.1 Dataset 01

For Study 1 and Study 2, we use quality-bitrate values for two video samples encoded using two different codecs. While the codecs and the values for bitrates and the two-objective metrics (PSNR and VMAF) used in this work are from actual measurements, for the purpose of this work we will refer to them as Codec A and Codec B (since our focus here is not on the comparison of codecs for their

compression efficiency gain, but rather to study the methods to do so). The Quality-BR curves for the two video samples considering both metrics and codecs are shown in Figure 1. From Figure 1, we can observe that for sample video 1, considering the PSNR-BR curve, the quality gap is low at the lower bitrate range, increasing at the higher bitrate range, while for VMAF the quality gap is almost the same at both low and high bitrate values. For sample Video 2, PSNR values are quite different (huge gap) while when considering VMAF values, the two curves almost overlap. Also, what is very interesting is that the performance of Codec B as compared to Codec A is worse when considering PSNR and better when considering VMAF. Hence, the RD curves are quite complex and show different behaviour for the compared codecs depending on the choice of the quality metric. A good BD-BR function (estimator of quality or bitrate savings metric) should be indicative of the observation from the respective RD curves.

2.2 Open-Source Implementations

We started with an evaluation of the open-source implementations summarized in Table 1. For easier access and reproducibility of the results, the consolidated list of these software and their implementation is provided also in our open-source dataset.

Table 1: Different open source implementations.

Implementation Name	Pseudonym	BD-BR Function	Software	Reference
ETRO's Bjontegaard Metric implementation	BD-BR_E	BD_BR_Improved	Excel	[15]
Bjontegaard_metric	BD-BR_P BD-BR_P_PW	BD_BR_Original BD_BR_Improved	Python	[2]
Bjontegaard metric calculation (BD-PSNR)	BD-BR_M	BD_BR_Original	MATLAB	[14]
BD-Rate/BD-PSNR Excel extensions	BD-BR_H	BD_BR_Original, BD_BR_Extended	Excel	[16]

- *Bjontegaard metric calculation (BD-PSNR)* [14] is a MATLAB implementation that supports BD-BR calculation with more than 4 data points. A third-order polynomial is fitted to the data using the *polyfit* function. The integration is calculated using the *polyint* function, following which the polynomial is evaluated for the integral values using the *polyval* function. This implementation is referred to as BD-BR_M in this work. Note: The Matlab code used here is an improved MATLAB version of the Bjontegaard metric [17] with correct integration intervals. The values obtained by the former implementation are not realistic and hence are not reported here.
- *Bjontegaard_metric* [2] is a Python implementation of the BD-BR_Improved function. The software provides the option to calculate the results using either the third-order polynomial curve fitting (referred to as BD-BR_P) or using the piecewise cubic polynomial interpolation (referred to as BD-BR_P_PW). However, in the implementation, the authors "fix" the case when the curve is not monotonic, by sorting the metric values. While this might not be an issue for most of the objective quality metrics, this might not be the actual representation of true RD curves when considering the MOS scores as the choice of quality metric, since MOS values, often at the higher end, are not always monotonic (as will also be evident in Study 3 using MOS scores as the quality metric).

  Note: There is also another Python implementation, *BD-metric 0.9.0* [10]. However, as the code base is the same, we use only the first implementation for the calculation of our results but both implementations should give the same results.
- *BD-Rate/BD-PSNR Excel extensions* are the Excel implementations provided by the proponents of the BD-BR_Extended function discussed earlier [16]. The default mode None is used for the newly proposed BD-BR_Extended function, shown in Table 2 as bdrateExtend and bdPSNRExtend respectively for BD-Rate and BD-Quality savings. The provided Excel sheet provides additionally the functions *bdrate* and *bdrateOld*. *bdrateOld* is the function implementing the BD-BR_Original function (with cubic interpolation) while *bdrate* is the BD-BR_Piecewise implementation, made originally available in [6]. The values computed by these two functions match exactly the results obtained from the implementations used in the Joint Video Experts Team (JVET) standardization work [5, 7]. Hence for brevity, we only report results from this implementation.

2.3 BD-Rate Calculation Using Various Implementations

As mentioned earlier, the original BD-BR metric, as well as the implementations used in standardization activities, considers only 4 DPs. However, recent works have reported BD-BR results considering more than 4DPs, with many open-source implementations supporting them. Hence, in this study we consider three different cases with respect to the number of data points (DPs) considered: All (considering all five DPs), Upper (top 4 DPs) and Lower (bottom 4 DPs). Based on the results presented in Table 2, the following can be concluded:

(1) All three implementations (BD-BR_M, BD-BR_E, and BD-BR_P) corresponding to the *BD-BR_Original* function provide similar results for all three different conditions, across both metrics and videos considered.
(2) *bdrateOld* values are the same as for other implementations of *BD-BR_Original* function when considering 4 DPs but differ when considering 5 DPs. Hence, in the BD-BR_H implementation, the old BD-BR functions (*bdrate* and *bdrateOld*) do not seem to support more than 4 DPs and result in different values as compared to the other three implementations. It should be recalled that these are the same implementations made available in [16] and also used in standardization activities by JVET, while the original implementation as well as the implementation currently used in standardization activities use only 4 DPs for BD-Rate computation.
(3) The results provided by the newly proposed metric, "BD-BR_Extended", used in the default mode are the same as those obtained by the BD-BR_P_PW (except for Video 2 using PSNR, where the savings figures appear to be undefined). This is because the *bdrateExtend* function in BD-BR_H implementation includes an additional form of error handling. However, looking at the respective RD curves, the value reported by BD-BR_P_PW implementation seems more realistic.

Table 2: BD-Rate savings results for implementations grouped by different quality metrics, PSNR and VMAF.

	PSNR											
	Video 1						Video 2					
	BD-BR_H		BD-BR_M	BD-BR_P	BD-BR_P_PW	BD-BR_E	BD-BR_H		BD-BR_M	BD-BR_P	BD-BR_P_PW	BD-BR_E
All (5 data points)	bdrate	151.45%	29.72%	29.72%	29.59%	29.72%	bdrate	289329.41%	175.23%	175.22%	183.41%	175.23%
	bdrateOld	27.78%					bdrateOld	237.09%				
	bdrateExtend	29.59%					bdrateExtend	183.40%				
Upper (top 4 DPs)	bdrate	44.79%	44.98%	44.98%	44.79%	44.98%	bdrate	0.00%	1933.86%	1933.86%	87.95%	1933.86%
	bdrateOld	44.98%					bdrateOld	1933.86%				
	bdrateExtend	44.79%					bdrateExtend	100.00%				
Lower (bottom 4 DPs)	bdrate	22.65%	22.65%	22.65%	22.65%	22.65%	bdrate	172.66%	171.54%	171.54%	172.66%	171.54%
	bdrateOld	22.65%					bdrateOld	171.54%				
	bdrateExtend	22.65%					bdrateExtend	172.66%				
	VMAF											
	Video 1						Video 2					
	BD-BR_H		BD-BR_M	BD-BR_P	BD-BR_P_PW	BD-BR_E	BD-BR_H		BD-BR_M	BD-BR_P	BD-BR_P_PW	BD-BR_E
All (5 data points)	bdrate	-74.76%	-21.70%	-21.69%	-21.70%	-21.70%	bdrate	-9.42%	-2.72%	-2.72%	-2.67%	-2.72%
	bdrateOld	-22.25%					bdrateOld	-14.21%				
	bdrateExtend	-21.70%					bdrateExtend	-2.67%				
Upper (top 4 DPs)	bdrate	-19.73%	-19.63%	-19.63%	-19.73%	-19.63%	bdrate	-3.20%	-1.50%	-1.50%	-3.20%	-1.50%
	bdrateOld	-19.63%					bdrateOld	-1.50%				
	bdrateExtend	-19.73%					bdrateExtend	-3.20%				
Lower (bottom 4 DPs)	bdrate	-23.09%	-23.10%	-23.10%	-23.09%	-23.10%	bdrate	-1.94%	-2.07%	-2.07%	-1.94%	-2.07%
	bdrateOld	-23.10%					bdrateOld	-2.07%				
	bdrateExtend	-23.09%					bdrateExtend	-1.94%				

(4) For *bdrateExtend*, considering 4DPs, for most cases the results of *BD-BR_Original* and *BD-BR_Improved* are almost equal. However, at the edge cases, the reported values (e.g., Video 3, VMAF) are very different. Considering the lower DPs, the values are more consistent.

2.4 Discussion

All three open-source implementations, BD-BR_E, BD-BR_M and BD-BR_P, for the considered cases seem to provide similar and reliable results and also can handle more than 4 DPs. However, it should be noted that BD-BR_P seems to support both third-order polynomial fit and piecewise cubic interpolation, with the results of BD-BR_P_PW closely resembling those obtained using the default BD-BR_Extended function. However, in some special cases, as discussed above, the values might differ a lot which indicates a possible difference between how excel and python functions handle different extreme cases. For the calculation of recommended BD-BR functions (BD-BR_Piecewise) for 4 or more than 4DPs, use the Python implementation in Piecewise mode or the excel implementation (BD-BR_H function bdrateExtend). For calculation of BD-BR_Piecewise considering 4DPs, either of the implementations above and *bdrate* function in BD-BR_H can be used.

3 STUDY 2: EVALUATION OF THE DIFFERENT BD-BR FUNCTIONS AND IMPLEMENTATIONS

Based on these results, we now limit the rest of our analysis to the following four implementations:

- BD-BR_H (BD-BR_Extended function in default mode which corresponds to the *BD-BR_Piecewise* function).
- BD-BR_M (*BD-BR_Original* function).
- BD-BR_P (*BD-BR_Original* function).
- BD-BR_P_PW (*BD-BR_Piecewise* function, python implementation).

3.1 Dataset 02

To check for the robustness of the metric's performance across a wide range of codecs as well as conditions, and also, in order to make sure that the results are generic and not restricted by the choice of codecs and videos used earlier, in Study 2 (and later in Study 3), we selected four video sequences encoded at 4 different bitrates using two of the most used encoders (H.264 and HEVC) from the open-source dataset available in [12]. Due to the difference in results observed when considering more than 4 DPs in Study 1, we restrict the number of DPs here to four. The Quality-Bitrate plots for all four videos considering four different metrics (PSNR, SSIM, VMAF and MOS) are not presented here for brevity but are made available in the open-source dataset [3]. The selection of the four videos was done based on their quality ratings range as well as considering whether MOS ratings were monotonically increasing with bitrate as explained next.

(1) Processed video sequences (PVSs) associated to Video 1: low-mid-high-quality range. MOS is non-monotonically increasing for Codec B.
(2) PVSs associated to Video 2: easy to encode, low-mid-high-quality range. MOS is monotonically increasing for both codecs.
(3) PVSs associated to Video 3: mid-high-quality range, MOS is non-monotonically increasing for Codec B.
(4) PVSs associated to Video 4: very hard to encode, lower quality range. MOS is monotonically increasing for both codecs.

It can be seen that the chosen videos and conditions are quite extreme cases and we believe that such "extreme" cases can help us

identifying more robust BD-BR functions and respective implementations.

Table 3: BD-Rate and BD-Quality results for four different videos considering three quality metrics.

	Video 1					
	BD-Rate			BD-Quality		
	PSNR	SSIM	VMAF	PSNR	SSIM	VMAF
BD-BR_E	-50.7%	-56.0%	-45.3%	2.72	0.05	13.20
BD-BR_M	-48.6%	-100.0%	-40.1%	2.61	0.05	13.01
BD-BR_P	-48.6%	79.0%	-40.1%	2.61	0.05	13.01
BD-BR-P_PW	-50.7%	nan	-45.3%	2.72	0.05	13.20
	Video 2					
	BD-Rate			BD-Quality		
	PSNR	SSIM	VMAF	PSNR	SSIM	VMAF
BD-BR_E	-28.0%	-34.9%	-25.4%	0.65	0.00	4.92
BD-BR_M	-27.9%	-100.0%	-69.5%	0.67	0.00	4.88
BD-BR_P	-27.9%	-35.1%	-69.5%	0.67	0.00	4.88
BD-BR-P_PW	-28.4%	nan	-25.4%	0.66	0.00	4.92
	Video 3					
	BD-Rate			BD-Quality		
	PSNR	SSIM	VMAF	PSNR	SSIM	VMAF
BD-BR_E	-50.8%	-53.9%	-47.0%	1.72	0.01	7.66
BD-BR_M	-41.7%	-100.0%	-38.2%	1.75	0.00	7.50
BD-BR_P	-41.7%	-56.1%	-38.2%	1.75	0.00	7.50
BD-BR-P_PW	-50.7%	nan	-47.0%	1.72	0.00	7.66
	Video 4					
	BD-Rate			BD-Quality		
	PSNR	SSIM	VMAF	PSNR	SSIM	VMAF
BD-BR_E	-33.6%	-39.3%	-12.4%	1.28	0.04	2.22
BD-BR_M	-32.1%	4.6%	-13.1%	1.20	0.03	2.51
BD-BR_P	-32.1%	4.6%	-13.1%	1.20	0.03	2.51
BD-BR-P_PW	-33.6%	-34.6%	-12.4%	1.27	0.04	2.22

3.2 BD-Rate and BD-Quality Results

In the absence of ground truth, it is difficult to estimate the "correct" BD-BR metric. However, we argue that a good and correct BD_BR metric should give ideally a good agreement over percentage bitrate savings (BD-Rate) and actual quality savings. Table 3 presents the BD-Rate and BD-Quality results for the four videos considering all three objective quality metrics separately. We will next discuss the results considering separately the BD-BR function, quality metric and videos.

The values provided by the implementation BD-BR_E and BD-BR_PW, as expected, are the same across all four videos for both BD-Rate and BD-Quality computation for all cases except for BD-Rate (SSIM). The values for BD-Rate, when considering SSIM as the quality metric, vary a lot, indicating unstable results. Similar observations hold true when considering the BD-BR_M and BD_BR_P implementations corresponding to the *BD-BR_Original* function.

Considerations on the results for the three quality metrics considered are reported below.

(1) PSNR: The results of the two BD-BR functions and their respective implementations' BD-Rate values vary with the values for Video 3 quite different from each other. However, corresponding BD-Quality savings are more consistent across all four videos for both functions.
(2) SSIM: Considering BD-Rate, the values reported for SSIM are quite different across both functions and implementations (except for Video 4, where, while the respective function implementation agrees, the difference between the *BD-BR_Original* and *BD-BR_Improved* are contradictory). However, we can see that the quality (SSIM) savings are in the order of 0.00-0.05. Hence, considering the BD-Quality (SSIM) savings values, as well as considering the RD plots, one can argue that the values reported by almost all BD-BR functions for BD-Rate (SSIM) are not realistic.
(3) VMAF: The results reported for BD-Rate savings by the two functions are quite different when considering VMAF as the quality metric, with the result for video 2 differing by a huge magnitude, and results for video 4 being the closest. When considering the BD-Quality (VMAF) savings, the agreement between the two functions is much better with the difference not being too high. Looking at the BD-Quality (VMAF) savings figures for Video 2, it can be argued that in this case, values reported for BD-Rate savings for BD-BR_Extended implementations seems to be more practical. However, a generalization, in this case, is hard to reach and further studies in this direction are required.

Based on our results, it is clear that all original BD-BR metrics and functions were designed considering only PSNR as the quality metric, and hence one should be careful when interpreting the BD-BR results obtained when using quality metrics other than PSNR, especially with SSIM.

4 STUDY 3: BD-RATE AND BD-QUALITY USING MOS

In addition to the two BD-BR implementations, for this study, we additionally consider the Subjective Comparison of ENcoders based on fItted Curves (SCENIC) metric [8], a metric proposed by Hanhart and Ebrahimi in [8] which computes the average bitrate and MOS difference between two RD curves considering subjective (MOS) scores. The basic argument behind the proposed metric is that since MOS is not a linear metric, a non-symmetrical function should be used to map bit rate values to MOS. A MATLAB based open-source implementation of the metric is available here. In the table below we report BD-Rate (MOS) and BD-Quality (MOS) results for the two BD-BR functions and their corresponding implementations and SCENIC (using the open source implementation available in [9]), for both rate and MOS savings.

From Table 4 above it can be seen that, for BD-Rate calculation, for all four videos, none of the three BD-BR functions reaches an agreement with quite different values, as well as, in some cases, indicating the contrasting performance of the codecs. The difference between BD-BR_P_PW and BD-BR_E, in this case, is primarily due to the fact that the former implementation sorts the MOS scores and hence the calculations are not representative of the actual RD curves. However, it is interesting to note that for the BD-MOS case,

Table 4: BD-Rate and BD-Quality (MOS).

	BD-Rate				BD-MOS			
	Video 1	Video 2	Video 3	Video 4	Video 1	Video 2	Video 3	Video 4
BD-BR_E	-44.39%	-2.17%	NaN	-10.90%	0.60	0.04	NaN	0.12
BD-BR_M	NaN	-99.97%	-100.00%	16.70%	0.76	0.18	0.71	-0.02
BD-BR_P	8.10%	-99.97%	13060843.60%	16.71%	0.76	0.18	0.71	-0.02
BD-BR_P_PW	71.36%	-1.57%	NaN	-11.12%	0.60	0.04	0.45	0.13
SCENIC	-59.53%	8.11%	-50.36%	-19.14%	0.80	-0.02	0.42	0.20

the video 1 values are all positive and even though not very similar, are not that different. This is surprising considering the case that Video 1 codec 2 had a non-monotonic behaviour for MOS scores for Codec 2. For SCENIC metric, considering the Quality(MOS)-Bitrate curves, the BD-Rate values does not seem to be very realistic, while the BD-MOS values does look more realistic. However, as also pointed out by the authors in their paper, in the absence of ground truth of actual bitrate or quality savings, one cannot truly quantify the performance/correctness of either of the metrics.

4.1 Discussion on suitability of metrics other than PSNR for BD-BR computation

Our results using two additional objective quality metrics: SSIM and VMAF indicates that the use of the BD-BR metric computation for metrics other than PSNR should be done with caution and that there is an opportunity for the development of additional BD-BR functions designed for use with metrics other than PSNR. The values of SSIM, given its highly non-linear nature with bitrate, at mid-to-high bitrate range, vary, by very small magnitudes. Hence, using SSIM as the quality metric for BD-BR calculations results in unreliable results, as observed in the results presented in Study 2 and 3. A possible alternative would be a change of scale to a linear scale or calculation of BD-BR values separately for different quality ranges, which remains for now a future work.

Additionally, one must consider the fact that one of the reasons behind using log(Bitrate) for BD-Rate calculation is that, otherwise, during the calculation of bitrate savings, a higher bitrate saving is obtained at the high bitrate end. The use of the log bitrate scale results in linear curves for the two quality-bitrate curves. The reciprocity of calculation of BD-Rate and BD-PSNR seems to work because in this case both PSNR and Rate are on the log scale. The question remains, however, with the use of quality metrics other than PSNR for BD-BR calculations, does the reciprocity remain valid? Another important factor to consider is that while the initial metric PSNR was unbounded, the other metrics such as SSIM, VMAF and MOS are not. As observed in our studies, and as also argued by the authors in [8], the saturation, especially at a higher bitrate range needs to be considered. This is more relevant now, considering that most of the services already target high quality ranges and very low and hence such a function can help one obtain more "practical" savings figures. Also, considering MOS as the quality metrics, Study 3 results indicates that when MOS scores are not monotonically increasing and/or when there is a cross over between the RD curves, the values obtained for BD-Rate and BD-Quality(MOS) can be very misleading (especially, if reported without mentioning the actual measurement values and RD curves).

5 RECOMMENDATIONS AND CONCLUSION

In this work, we evaluated various implementations of the BD-BR function. We found that depending on the implementation used, for the same dataset, different values can be obtained. This is primarily due to the use of the since "deprecated" *BD-BR_Original* function instead of the recommended *BD-BR_Piecewise* function. Also, some implementations support only 4 DPs and hence should not be used for RD curves consisting of 5 or more DPs. Therefore, it is recommended that any future publications using BD-BR computation should provide a reference to their implementation, discuss the functions and formula used or use one of the recommended open-source implementations (see section 2.4).

While Gisle Bjontegaard in [4] initially did not find much benefit from distinguishing between "mid-range" and "total-range", an improvement submitted in 2008 by the same author included an implementation to calculate "high" and "low" calculations using a second-order polynomial fitted through three points. More recent studies by Netflix for large scale code comparison with eight data points consider three different quality ranges: low, medium and high (depending respectively on where the quality values correspond to the bottom four, middle four and top four quality values of the RD curve. Their results indicate that, depending on the selected quality range, the bitrate saving can differ a lot. Our results also indicated slightly more "stable" results and agreement between different implementations, when calculating the BD-BR results at the lower bitrate range, compared to the higher bitrate ranges. However, this is not fully conclusive and more analysis is required in this direction, which we leave for future work.

When using quality metrics other than PSNR, our results also showed that metrics such as SSIM and MOS might result in very contradicting results, with values that are often not realistic. None of the evaluated BD-BR functions always agree on actual bitrate savings; however, in terms of quality savings, most of them converge and agree to a certain extent. Considering MOS, due to its possible non-monotonic nature, all metrics provide different results and hence it is challenging to agree on a particular "savings" figure, be it bitrate or MOS. Hence, BD-BR results for such metrics should be calculated and reported with caution. Also, there exist some opportunities and challenges on the design of better BD-BR metrics when considering quality metrics other than PSNR.

In short, unless the RD curves for the two codecs compared are "well behaved", BD-BR results should be interpreted with caution and supported with additional measurements such as BD-Quality savings figures and RD plots. Also, as discussed earlier, if considering RD curves with more than 4 DPs, the values can be divided into different quality ranges such as high, medium, and low. Ideally, the bitrate values chosen should be representative of the real-world operating points (depending on the application). Hence, the reported figures should be used with caution as many studies have found contrasting results for codec compression efficiency when considering different encoding settings. Our future work will focus on understanding of the usefulness/distributions of different bitrates or quality ranges considering different applications and possible replacement/supplementation of BD-BR results with weighted PSNR/SSIM/VMAF averages considering such distributions.

REFERENCES

[1] Kenneth Andersson, Rickard Sjöberg, and Andrey Norkin. 2009. Reliability metric for BD measurements. ITU-T SG16 Q.6 Document, VCEG-AL22.

[2] Anserw. 2016. Bjontegaard_metric. https://github.com/Anserw/Bjontegaard_metric.

[3] Nabajeet Barman, Maria Martini, and Yuriy Reznik. 2021. Bjontegaard Delta Bitrate (BD-BR). https://github.com/NabajeetBarman/BD-BR.

[4] Gisle Bjøntegaard. 2001. Calculation of average PSNR differences between RD-curves. ITU-T SG16/Q6 input document VCEG-M33..

[5] Frank Bossen. 2011. Common HM test conditions and software reference configurations. Joint Collaborative Team on Video Coding (JCT-VC) of ISO/IEC MPEG and ITU-T VCEG document JCTVC-G1200.

[6] Frank Bossen. 2011. Excel template for BD-rate calculation based on Piece-wise Cubic Interpolation. hm32piecewisecubic2.xls.

[7] Frank Bossen, Xiang Li, Andrey Norkin, and Karsten Sühring. 2019. JVET AHG report: Test model software development (AHG3). Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 document JVET-O0003.

[8] Philippe Hanhart and Touradj Ebrahimi. 2014. Calculation of average coding efficiency based on subjective quality scores. *Journal of Visual Communication and Image Representation* 25, 3 (2014), 555–564. https://doi.org/10.1016/j.jvcir.2013.11.008 QoE in 2D/3D Video Systems.

[9] Philippe Hanhart and Touradj Ebrahimi. 2014. SCENIC. https://github.com/phanhart/SCENIC.

[10] Shengbin Meng. 2019. Bjontegaard Metric. https://pypi.org/project/bd-metric/.

[11] ITU-T Technical Paper. 2020. HSTP-VID-WPOM - Working practices using objective metrics for evaluation of video coding efficiency experiments.

[12] Rakesh Rao Ramachandra Rao, Steve Göring, Werner Robitza, Bernhard Feiten, and Alexander Raake. 2019. AVT-VQDB-UHD-1: A Large Scale Video Quality Database for UHD-1. In *2019 IEEE ISM*. 1–8.

[13] ITU-T Recommendation. 2020. ITU-T H.266 (08/2020) Versatile video coding. https://www.itu.int/itu-t/recommendations/rec.aspx?rec=14336.

[14] Serge. 2021. Bjontegaard metric calculation (BD-PSNR). https://www.mathworks.com/matlabcentral/fileexchange/41749-bjontegaard-metric-calculation-bd-psnr.

[15] ETRO Tim Bruylants. 2016. ETRO's Bjontegaard Metric implementation. https://github.com/tbr/bjontegaard_etro.

[16] Alexis M. Tourapis, David Singer, Yeping Su, and Khaled Mammou. 2017. BD-Rate/BD-PSNR Excel extensions. Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11.

[17] Giuseppe Valenzise. 2021. Bjontegaard metric. https://www.mathworks.com/matlabcentral/fileexchange/27798-bjontegaard-metric.