2018 IEEE International Symposium on Multimedia (ISM)

# Light-weight Video Coding Based on Perceptual Video Quality for Live Streaming

Yusuke Sakamoto, Shintaro Saika, Masaru
Takeuchi, Tatsuya Nagashima, Zhengxue Cheng,
Kenji Kanai and Jiro Katto
Dept. of Computer Science and Communications
Engineering
Waseda University
Shinjuku-ku, Tokyo, Japan
{y_sakamoto, katto}@katto.comm.waseda.ac.jp

Kaijin Wei, Ju Zengwei and Xu Wei
Huawei Technologies
Longgang District, Shenzhen, China
{weikaijin1, juzengwei, xuwei35}@huawei.com

*Abstract*— **In video streaming on the internet, effective encoding recipes (i.e. bitrate-resolution pairs) are a main obstacle to deliver high-quality video streams. We developed a method to generate an encoding recipe that considers subjective visual quality with one just-noticeable difference (JND) distance. However, this method requires excessive computation time, which is not directly applicable for live streaming. In this paper, in order to provide a light-weight method for live streaming, we developed three acceleration techniques: resolution extrapolation, VMAF skipping and sampled objective measure calculation. These techniques are heuristic, but greatly contribute to reducing computational cost. Experimental results demonstrate that the proposed method achieves a significant reduction in computation time without significant effects on rate-JND characteristics.**

***Keywords- Adaptive video coding, computation time, JND, live streaming, perceptual video quality***

## I. INTRODUCTION

In recent decades, video streaming services have grown rapidly and become popular network services. In addition, users' viewing environments have been diverse owing to the evolution of mobile devices. Thus, people can enjoy video contents anytime and anywhere. Therefore, generating an "encoding recipe" (i.e., selecting bitrates and resolutions), is important to provide high-quality video streaming services.

According to Netflix's technical blog, an empirical fixed encoding recipe had been utilized as shown in the left side of Table I [1]. This table, which we call "fixed recipe," is defined as the fixed pairs of bitrates and resolutions, regardless of video sequences. Per-title encode optimization proposed by Netflix tries to provide an encoding recipe in an adaptive manner to input video characteristics [1]. However, their publicly available methods [3,4] gave an encoding recipe which is optimized from the classical rate-distortion viewpoint, and introduction of perceptual video quality into the encoding recipe design can be considered.

To address this fact, we have developed a perceptual quality-driven video coding method for video on demand (VOD) streaming [2]. Our method provides a machine-learning-based perceptual quality (JND) estimator that quantifies a relationship between perceptual quality and encoding rates, and an optimized encoding recipe is generated by pre-analyzing the source video and applying the pre-trained JND estimator. Although this system can achieve higher perceptual quality and lower storage cost than the fixed recipe, its computational cost is not appealing. This is because computational cost of pre-analysis step is quite high.

Therefore, in this paper, we propose a heuristic but light-weight video coding method which can be applied to live streaming. Our contributions can be summarized by two aspects. Firstly, we propose three techniques to reduce computational cost as follows: 1) we estimate rate-distortion curves of high resolution encoding results from those of low resolution results (called "resolution extrapolation"), 2) we skip computationally heavy VMAF [6] calculation in the pre-analysis by sacrificing JND estimation performance (called "VMAF skipping"), and 3) we conduct temporally sampled calculation of objective quality measures (called "sampled objective measure calculation"). Secondly, we validate the performance of the proposed method by using 220 test sequences of VideoSet [7] and confirm that the proposed method can greatly reduce computational complexity while maintaining acceptable coding efficiency.

## II. RELATED WORK

### A. *Netflix's Encoding Recipe Selector*

Netflix has proposed an encoding recipe selection method that pre-analyzes the complexity of source video [3]. In their proposal, the system first divides the source video into $N$ segments and extracts 1-minute video frames from the center of each segment as a sample data. Following this, each

TABLE I. EXAMPLE OF ENCODING RECIPES.

| | **Fixed recipe [1]** | | **CJND recipe [2]** | | |
|---|---|---|---|---|---|
| **Level** | ***Resolution*** | ***Bitrate [kbps]*** | ***QP*** | ***Resolution*** | ***Bitrate [kbps]*** |
| 1 | 1920×1080 | 5800 | 29 | 1920×1080 | 1154.06 |
| 2 | 1920×1080 | 4300 | 27 | 1280×720 | 592.358 |
| 3 | 1280×720 | 3000 | 31 | 1280×720 | 366.442 |
| 4 | 1280×720 | 2350 | 34 | 1280×720 | 262.726 |
| 5 | 720×480 | 1750 | 36 | 1280×720 | 212.405 |
| 6 | 720×480 | 1050 | 38 | 1280×720 | 172.622 |
| 7 | 512×384 | 750 | 36 | 960×540 | 146.823 |
| 8 | 512×384 | 560 | 32 | 640×360 | 121.343 |
| 9 | 384×288 | 375 | 34 | 640×360 | 98.5214 |
| 10 | 320×240 | 235 | 37 | 640×360 | 72.5084 |

978-1-5386-6857-3/18/$31.00 ©2018 IEEE
DOI 10.1109/ISM.2018.00033

extracted sample is encoded by multiple fixed quantization parameters (QPs) and the bitrates and peak signal-to-noise ratios (PSNRs) are calculated from the encoded video samples. Next, a fitted rate-PSNR curve is determined from the average bitrates and PSNRs in each QP. Finally, from the rate-PSNR curve, the best encoding recipe is selected from pre-sets by the bitrate that achieves a certain PSNR threshold.

### *B. Netflix's Per-Title Encoding Optimization*

Likewise, Netflix has also proposed the per-title encoding method [4]. Moreover, to optimize the encoding recipe, they proposed two methods: per-title complexity analysis and per-chunk bitrate control.

In per-title complexity analysis, there are three steps: 1) trial encoding, 2) curve estimation, and 3) encoding recipe generation. In the first step, whole frames of a source video are encoded by different constant rate factor (CRF) values for multiple resolutions, and the bitrates and quality of encoded videos are then calculated in each trial encoding. Second, the R-D curve for each resolution is estimated from the observed bitrates and quality metrics by applying curve interpolation. Finally, an encoding recipe is derived by selecting the R-D samples at the closest point of the convex hull.

Per-chunk bitrate control provides target resolution-bitrate pairs *($R_i$, $B_i$)* for each video chunk by applying multi-pass encoding. A source video is divided into fixed-size chunks, and an encoding process is performed independently for each chunk *n*. In the first pass encoding, a video chunk is encoded by the CRF value $C_i$ that corresponds to a pair of *($R_i$ $B_i$)*. It should be noted that these $C_i$, $R_i$, and $B_i$ values can be obtained by per-title complexity analysis. After the first pass encoding at $C_i$, the bitrate $B_{i,n}$ is calculated. When the $B_{i,n}$ is smaller than $B_i$, the second pass encoding (with the target rate $B_{i,n}$) is performed by using per-frame statistics derived from the first pass encoding. When $B_{i,n}$ is larger than $B_i$, a regular two-pass encoding with the target rate $B_i$ is performed.

### *C. Constant JND Recipe*

Inspired by this research, the authors have developed an adaptive encoding recipe generation method based on perceptual quality [2], where JND [5] is adopted as a perceptual quality metric. The overview of this method, called Constant JND (CJND) recipe generation, is shown in Fig. 1. This method is mainly composed of four steps: 1) pre-analysis, 2) curve fitting, 3) JND estimation, and 4) recipe generation.

First, source video is pre-encoded using three QPs (15, 30 and 45) and four resolutions (1920×1080, 1280×720, 960×540 and 640×360). Three types of encoded distortion measures (PSNR, SSIM, and VMAF [6]) are then calculated (in the pre-analysis step). Second, based on the results of the pre-analysis step, the bitrate and distortion scores for QPs 0-51 are estimated by fitting QP-bitrate and QP-distortion curves on each resolution (the curve fitting step). The approximation function we used is:

$$y = a \cdot x^b + c \tag{1}$$

where *x* indicates QP, and *y* indicates bitrate, PSNR, SSIM, or VMAF. Third, by using these results as inputs, JND scores are estimated by applying a support vector regression (SVR) model (the JND estimation step). To train this SVR model, we used ground truth JND values of VideoSet and subjective evaluation experiments carried out by the authors. It should be noted that these steps are performed per resolution; thus, pairs of bitrate and JND scores are obtained for corresponding QP values for each resolution. The encoding recipe is then generated, in which neighboring levels have one JND distance.

Table I (right side) shows an example of a CJND encoding recipe. Compared to fixed recipes, CJND recipes can save bitrates by guaranteeing perceptual quality. However, the recipe generation requires huge computational resources because of the complexity of the pre-analysis step. Thus, in [2], this recipe only applies into VOD streaming.

## III. Acceleration of Constant JND Recipe Generation

To reduce the computational cost of CJND recipe generation and apply it to live streaming, we provide a light-weight pre-analysis step that is composed of three techniques. The concept of live streaming extension is also shown in Fig. 1 (right side: Live); the details are described below.

### *A. Resolution Extrapolation*

The main purpose of resolution extrapolation is to estimate the pre-encoding results (i.e., QP-bitrate and QP-distortion curves) for higher resolutions from that of lower resolutions. In [2], the pre-analysis step consumes the highest computation time as these steps are conducted for all resolutions (e.g., 4 resolutions = 4 pre-analyses), as shown in Fig. 1 (left side: VOD). Thus, skipping the pre-analysis step for some resolutions can greatly contribute to a reduction in computation time.

To estimate the QP-bitrates and QP-distortion curves, the curve fitting parameters *a, b,* and *c* in Equation (1) must also be estimated. An example of the distribution of parameters *a, b*, and *c* for four different resolutions are shown in Fig. 2, the results of which potentially indicate that the parameters for high resolutions (1080p and 720p) can be estimated from

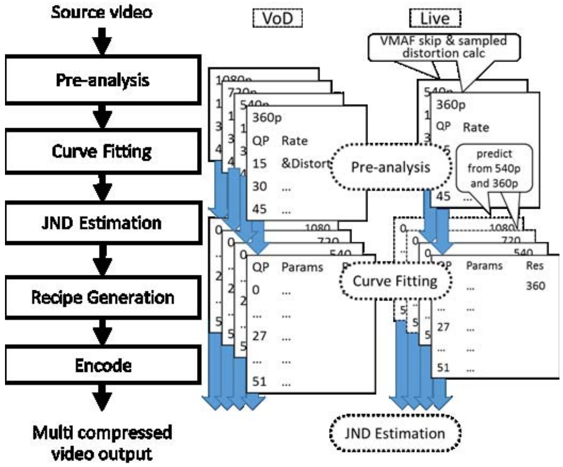
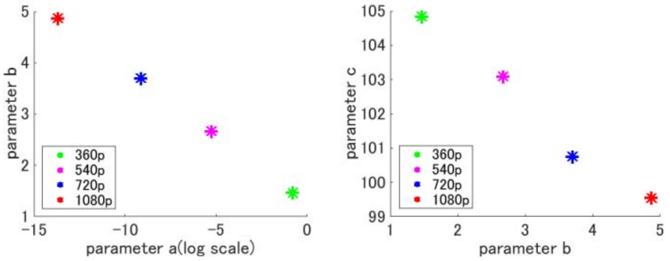


Figure 1. Overview of constant JND recipe generation.



(a)parameter pair (a, b)

(b) parameter pair (b, c)

Figure. 2 An example of the distribution of parameter pairs (a, b) and (a, c) in Equation (1).

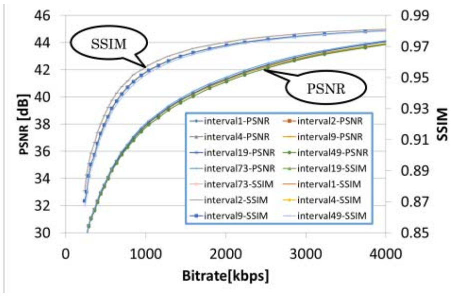


Figure. 3 Rate-PSNR curves in different numbers of skipping frames.

those of low resolutions (540p and 360p). This is because the parameters *(a, b, c)* of QP-bitrates and QP-distortion curves tend to be on the same line at regular intervals according to their individual resolutions. Therefore, the parameters *(a, b, c)* of high resolutions (1080p and 720p) are calculated by using Equations (2)-(4) (i.e., resolution extrapolation):

$$\log(a_r) = \log(a_{540}) + k\{\log(a_{540}) - \log(a_{360})\} \quad (2)$$

$$p_r = p_{540} + k(p_{540} - p_{360}) \quad (3)$$

$$k = \begin{cases} 1, & r = 720 \\ 2, & r = 1080 \end{cases} \quad (4)$$

where *p* indicates the parameter *b* or *c* in Equation (1), and 1080, 720, 540, and 360 indicate the resolutions.

Although some sequences do not follow this characteristic, we can confirm that several sequences do follow it by testing using the VideoSet [7] database. Consequently, the pre-analysis step of high resolutions (1080p and 720p) can be skipped, which represents a reduction in computation cost.

### *B. VMAF Skipping*

In addition, it is true that skipping the calculation of distortion metrics can contribute to reduce computational cost of pre-analysis steps. Although the accuracy of the JND estimator will be improved by using larger numbers of video features, the computation cost for distortion measures will be drastically increased. Thus, we compounded such trade-off characteristics and skipped the VMAF calculation that requires the highest computation cost (Table II). In this case, we re-trained the JND estimator by using PSNR and SSIM. However, the accuracy of JND estimator unsurprisingly decreased compared with the VMAF usage case (Table III) further emphasizing the effects of cost reduction.

### *C. Sampled Calculation of Distortion Measures*

Natural video sequences contain significant temporal redundancy. Therefore, distortion measurements were performed using sampled frames (i.e., skipping some frames) to avoid further computation costs in the pre-analysis step. As shown in Fig. 3, the results of R-D curves for different numbers of skipping frame(s) (1 to 73) are similar. In addition, Table IV shows comparisons of computation time for different number of skipping frame(s). Although some sequences do not accurately fit the R-D curve, we have confirmed that several sequences work well. Thus, the sampled calculation of distortion measures may reduce computation time in the pre-analysis step without a failure of R-D curve fitting.

## IV. Performance Evaluations

### *A. Experimental Conditions*

We evaluated a trade-off characteristic (i.e., total required time and reliability of encoding recipe) of the proposed acceleration methodology using 220 video sequences (from the VideoSet database [7]) and validated performance by comparing the fixed recipe generation of Netflix [1] with the original CJND recipe generation (VOD) in [2]. The total required time was calculated by completing all process steps, (pre-analysis, recipe generation, and encoding). In addition, to evaluate the reliability of the encoding recipe, we applied the Bjontegaard delta metric [8] to the rate-JND curve and calculated a BD-rate against the fixed recipe. Evaluation Environment is a public cloud server provided by SAKURA Internet [9], which is one of the most popular cloud providers in Japan. It should be noted that the resolution and frame rates of the source video were 1920×1080 and 30/24 fps, respectively; the duration time was five seconds.

TABLE I. Computation Time Results of Objective Distortion Measures in Total 150 Frames.

| | Metric | | |
|---|---|---|---|
| | ***PSNR*** | ***SSIM*** | ***VMAF*** |
| Process time (sec) | 1.39 | 21.54 | 97.39 |

TABLE II. Estimation Errors of JND Estimation Using Different Parameters. (MSE: Mean Squared Error)

| **Parameters** | | **MSE** |
|---|---|---|
| **Resolutions, QPs, bitrates (common parameters)** | | 0.822 |
| | **PSNR** | 0.798 |
| | **PSNR + SSIM** | 0.727 |
| | **PSNR + SSIM + VMAF** | 0.402 |

TABLE III. Comparison of Processing Times of PSNR and SSIM Measurements.

| | Number of skipping frame(s) | | | | | | |
|---|---|---|---|---|---|---|---|
| | ***1*** | ***2*** | ***3*** | ***9*** | ***19*** | ***49*** | ***73*** |
| Process time (sec) | 8.1 | 5.4 | 3.2 | 1.6 | 0.9 | 0.4 | 0.4 |

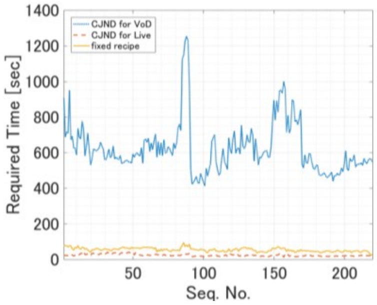


Figure. 4 Required time in three recipe generation methods for 220 video sequences.

### B. Experimental Results

First, Fig. 4 illustrates the required time for three recipe generation methods from 220 video sequences; we confirmed that the acceleration methodology can achieve 96.2 % computational cost reduction on average. In addition, the proposed method is 56.2% faster than the fixed recipe approach (no recipe generation processes). This is mainly because the CJND for live streaming selects lower bitrates and resolutions for the encoding recipe.

Fig 5 demonstrates the comparison results of BD-rate against the fixed recipe among 220 video sequences. Thus, results show that most sequences can achieve more efficient encoding recipes in both CJND for VOD and live streaming, which can 34.5% and 32.4% average bit reduction respectively. In addition, although some live streaming sequences in CJND were less efficient than VOD, this acceleration methodology can cause suppression, which leads to efficiency degradation of encoding recipe owing to the reduction of computational complexity.

Finally, Fig. 6 shows the relationships between actual JND and target JND for two CJND recipes: VOD, and live streaming. Actual JND was re-calculated JND from encoded video; target JND was equal to the "Level" value in Table I. It should be noted that this value represents the JND for CJND recipes. As shown in the Fig.6, the CJND recipes for live streaming roughly indicate a similar characteristic to VOD streaming and approximately generate accurate JNDs (i.e., actual JND is equal to target JND). Thus, these results conclude that the proposed acceleration methodology achieves a significant reduction in computation time without a large degradation in the reliability of the encoding recipe.

## V. Conclusion and Future Work

In this paper, we proposed an acceleration methodology to generate perceptual quality-driven encoding recipes for live streaming. Our previous method [2] can optimize perceptual quality in a JND sense, but suffers from excessive computational time in the pre-analysis step. We therefore introduced three acceleration methods; resolution extrapolation, VMAF skipping, and sampled calculation of distortion measures. Through evaluations using 220 video sequences, our experimental results demonstrated that the proposed method can reduce computational cost without a large degradation in the rate-distortion performance. In future work, we will provide deeper analysis and discussion on the evaluation results and improve the proposed method to generate more accurate constant JND recipes with low computation cost. In addition, we will implement the proposed method into an actual platform and validate the perceptual quality by carrying out actual live streaming experiments along with greater numbers of subjective evaluations.

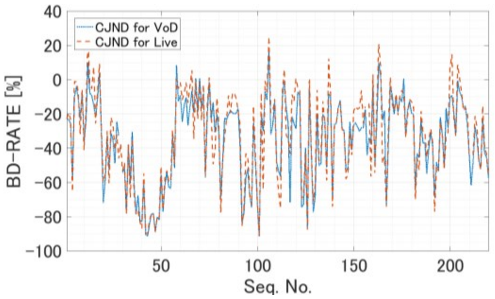
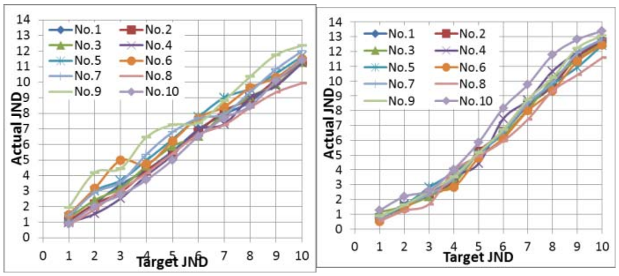


Figure. 5 The comparison of BD-rate with fixed recipes from 220 video sequences. (Lower values represent higher coding efficiency, and vice versa.)



(a)CJND for VOD (b)CJND for Live

Figure. 6 Relationships between actual JND and target JND (Due to limitations of space, only 10 sequences (No. 1~10.) are shown.)

## References

[1] A. Aaron, Z. Li, M. Manohara, J. D. Cock and D. Ronca, (2015) “Netflix Tech Blog: Per-Title Encode Optimization,” [Online]. Available at: http://techblog.netflix.com/2015/12/per-title-encode-optimization.html.

[2] M.Takeuchi, S. Saika, Y. Sakamoto, T. Nagashima, Z. Cheng, K. Kanai, J. Katto, K. Wei, J. Zengwei and X. Wei, "Perceptual Quality Driven Adaptive Video Coding Using JND Estimation" in *PCS 2018*, June 2018.

[3] A. Aaron, D. Ronca, I. Katsavounidis and A. Schuler, (2016) “WO 2016160295 A1: Techniques for optimizing bitrates and resolutions during encoding,” [Online]. Available at: https://www.google.com/patents/WO2016160295A1.

[4] J. D. Cock, Z. Li, M. Manohara and A. Aaron, “Complexity-based consistent-quality encoding in the cloud,” in *IEEE ICIP 2016*, Sep.2016.

[5] Video Clarity, “Understanding MOS, JND, and PSNR,” [Online]. Available at: http://videoclarity.com/PDF/WPUnderstanding JNDMOSPSNR.pdf. (viewed at Jan. 28 2018.)

[6] Z. Liu, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, “Toward A Practical Perceptual Video Quality Metric,” Netflix Tech. Blog. [Online]. Available at: https://medium.com/netflix-techblog/toward-a-practical-perceptual-video-quality-metric-653f208b9652.

[7] H. Wang, I. Katsavounidis, J. Zhou, J. Park, S. Lei, X. Zhou, M.-On Pun, X. Jin, R. Wang, X. Wang, Y. Zhang, J. Huang, S. Kwong, and C.-C. Jay Kuo, “VideoSet: A Large-Scale Compressed Video Quality Dataset Based on JND Measurement,” *Journal of Visual Communication and Image Representation*, vol 46, pp 292-302, Jul. 2017.

[8] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves,” ITU-T VCEG-M33, 2011.

[9] (2018) SAKURA Internet, [Online]. Available at: https://cloud.sakura.ad.jp/. (in Japanese)