FAST BLOCK STRUCTURE DETERMINATION IN AV1-BASED MULTIPLE RESOLUTIONS VIDEO ENCODING

Bichuan Guo*, Yuxing Han†, Jiangtao Wen*

*Tsinghua University, †South China Agricultural University
jtwen@tsinghua.edu.cn

ABSTRACT
The widely used adaptive HTTP streaming requires an efficient algorithm to encode the same video to different resolutions. In this paper, we propose a fast block structure determination algorithm based on the AV1 codec that accelerates high resolution encoding, which is the bottle-neck of multiple resolutions encoding. The block structure similarity across resolutions is modeled by the fineness of frame detail and scale of object motions, this enables us to accelerate high resolution encoding based on low resolution encoding results. The average depth of a block’s co-located neighborhood is used to decide early termination in the RDO process. Encoding results show that our proposed algorithm reduces encoding time by 30.1%-36.8%, while keeping BD-rate low at 0.71%-1.04%. Comparing to the state-of-the-art, our method halves performance loss without sacrificing time savings.

Index Terms— Adaptive HTTP streaming, multiple resolutions, fast encoding, AV1

1. INTRODUCTION
Adaptive HTTP streaming has now been widely used by most video content providers to improve the quality of experience (QoE) of their users [1]. Videos are stored as multiple representations with varying sizes and qualities, and the client-side player requests a suitable representation according to the network condition [2]. The video spatial resolution, being one of the greatest factors affecting the video bit-rate, is used by multiple popular video-sharing websites (e.g. Youtube, Twitch) as the primary option to control the video quality. Therefore, a fast algorithm that encodes the same video to different resolutions is certainly of great interest.

AOMedia Video 1 (AV1) [3] is an emerging video codec developed by the Alliance for Open Media, which is open-source and royalty-free. Comparing to its predecessor VP9, it offers numerous new coding tools to achieve a cutting edge coding efficiency, at the expense of complexity. As a result, fast encoding is very challenging especially regarding to adaptive HTTP streaming, where the same video needs to be encoded for multiple times.

To speed up the encoding, one needs to exploit the correlation between multiple rate distortion optimizations (RDO). The block structure is mainly determined by the fineness of frame detail, and the amount of inter-frame motion. Therefore, the block structure preserves to a large extent during target resolution rescaling [4]. Motion vectors can also be reused, as they represent the motion of objects and therefore shall be consistent across different resolutions up to a rescale factor. However, an early termination algorithm that prevents unnecessary block structure search would also terminate all subsequent motion estimations, hence it is not surprising that the encoding complexity can be greatly reduced by solely considering block structures, as proven by several papers [5][6].

In this paper, we propose a fast block structure determination algorithm for AV1-based multiple resolutions encoding. Encoding processes with high target resolutions are accelerated by referring to a low target resolution encoding process to infer block structures and execute RDO early termination accordingly. Based on our statistical analysis, the average block depth of the co-located neighborhood, coupled with adaptive search range and threshold value, yield early termination decisions according to two distinct strategies designed for different block sizes.

The rest of the paper is organized as follows. Related work is presented in Section 2. The block structure of AV1 and the block structure similarity across resolutions are studied in Section 3. Section 4 describes how to exploit this cross-resolution similarity to gain information about block structures based on low resolution RDO results, and in turn accelerates high resolution encoding. Experimental results are given in Section 5, and Section 6 concludes the paper.

2. RELATED WORK
There are many frameworks dedicated to reduce the overall complexity of multiple representations video encoding. Transcoding methods [7][8] reduce the complexity by reusing RDO mode decisions and motion estimation results, obtaining a new representation through residual re-quantization.
from an existing bit-stream. However this introduces significant quality losses due to re-quantization, and it is not suitable for multiple resolutions encoding. Scalable video coding \cite{9} implements multiple layers in a single bitstream corresponding to different qualities, so that users can adaptively request the suitable sub-stream. While its performance is better than transcoding, it is still not desirable comparing to single layer coding, especially when the number of layers is high, and where bandwidth resource is limited.

The idea of fast multiple representations encoding was first introduced in \cite{10}. The RDO redundancy among different encoding processes is examined in order to gain speedup without introducing substantial rate-distortion (RD) loss. \cite{11} proposed a preliminary framework for same resolution, multiple target bit-rates encoding, where RDO decisions from the highest bit-rate encoding process were copied to other processes. This method resulted in considerable RD loss and was later improved by an RDO tree pruning algorithm \cite{4}\cite{5}. \cite{12} proposed a heuristic multiple resolutions encoding algorithm for HEVC, where low resolution encoding is accelerated by a high resolution reference encoding. However, multiple resolutions encoding are often done in parallel, and the highest resolution takes the longest time. Therefore, the aforementioned approach does not offer much benefit in practical situations.

As its main contribution, this paper proposes a fast block structure determination algorithm for AV1 that accelerates high resolution encoding based on low resolution encoding results. It has perfect parallel compatibility, in a sense that when all representations are encoded simultaneously, the time reduction on high resolution encoding fully translate into overall time reduction. Unlike most empirical algorithms listed above, the block structure similarity across resolutions is modeled by the fineness of frame detail and scale of object motions, the determination algorithm is then derived from statistical hypothesis testing.

### 3. BLOCK STRUCTURE SIMILARITY

#### 3.1. Block structures in AV1

Similar to HEVC, AV1 also uses a quadtree-based block structure. Each frame is partitioned into $64 \times 64$ blocks, and each square block can be further partitioned in a recursive manner. To be specific, a $2N \times 2N$ block can be partitioned into four $N \times N$ blocks (4-way split), two $N \times 2N$ blocks (2-way horizontal split), two $2N \times N$ blocks (2-way vertical split), or remain unpartitioned (see Fig. 1). The smallest block size is $4 \times 4$, and non-square blocks cannot be further partitioned. There are experimental tools in AV1 that support $2 \times 2$ blocks and the so-called “T-split”, however they will not be discussed in this paper as they are not included in the standard settings.

As there are square and non-square blocks, our follow-

![Fig. 1. Four ways to partition a square block in AV1.](image)

Fig. 1. Four ways to partition a square block in AV1.

As there are square and non-square blocks, our follow-

![Fig. 2. Block structure of the 1st frame of the BasketballDrive sequence, QP 27.](image)

Fig. 2. Block structure of the 1st frame of the BasketballDrive sequence, QP 27.

#### 3.2. Cross-resolution similarity

To observe the block structure similarity among different resolutions, the BasketballDrive test sequence from CTC \cite{13} is encoded using the AV1 codec to target resolutions $1920 \times 1080$ and $1440 \times 810$, both using a constant QP of 27. Fig. 2 shows the block structure of the first frame. It is clear that finely partitioned areas are in common: the athletes and the ceiling, i.e. areas with high level detail or large motion. This inspires us to use the fineness of frame detail and the
scale of object motions to model the block structure similarity across resolutions.

Therefore, an assumption is made that there exists a continuous function \( f \) defined on the frame plane which represents the fineness of frame detail and scale of object motions at each point. The value of this function is inherent to the scene, and does not depend on the target resolution of the encoder. The closed form of \( f \) will most likely depend on the image gradient and the optical flow, however, our following discussion only relies on the existence of such a resolution-invariant continuous function.

Denote the high (low) resolution encoding process by \( P_1 \) (\( P_2 \)). Suppose a block \( B_1 \) with depth \( d \) in \( P_1 \) has a neighboring block \( B_2 \) with depth \( d \) in \( P_2 \) (that is, they are close to each other). Let \( X_i \) denote the partition choice of \( B_i \), such that \( X_i = 1 \) when \( B_i \) is 4-way split, otherwise (non-split or 2-way splits) \( X_i = 0 \). Our previous observation states that \( X_i \) tends to be 1 where \( f \) takes large values, thus an assumption is made that the partition choice of a block is decided by its size and the value of \( f \) in its area. Formally, denote the average value of \( f \) in \( B_i \) as \( \mu_{B_i} \), \( \sigma_{X_i} \) can be modeled as a Bernoulli random variable with \( P(X_i = 1) = g_i(\mu_{B_i}) \), where \( g_i \) is a monotonically increasing function, representing the positive correlation between the tendency of any block in \( P_1 \) with depth \( d \) being partitioned, and the value of \( f \) in the block’s area. The assumption of \( f \) being resolution-invariant and continuous gives the following relation:

\[
EX_i = g_i(\mu_{B_i}) \approx g_i(\mu_{B_2}) = g_1 \circ g_2^{-1}(EX_2). \quad (1)
\]

The approximation is due to \( B_1 \) being close to \( B_2 \), therefore the average values of \( f \) are also close. It then follows that \( g_1 \circ g_2^{-1} \), the composition of \( g_1 \) and the inverse of \( g_2 \), is also increasing. This explains our intuition that wherever blocks in \( P_2 \) are finely partitioned, so are those in \( P_1 \).

However, there are many cases where blocks in Fig. 2(b) are not partitioned, but their neighbor blocks in Fig. 2(a) are. This inconsistency results from the variance of \( X_i \). In fact, the block structure similarity is more consistent in the average sense. Suppose \( B_i \) has a neighbor of blocks \( B_{ij} \) \( (1 \leq j \leq n) \) in \( P_1 \) with the same size as \( B_i \), relate \( X_{ij} \) to \( B_{ij} \) as \( X_i \) to \( B_i \), define

\[
X'_i = \frac{1}{n} \sum_j X_{ij}. \quad (2)
\]

Assuming independence between \( X_{ij} \), we have

\[
EX'_i = \frac{1}{n} \sum_{j=1}^n g_i(\mu_{B_{ij}}), \quad (3)
\]

\[
\sigma X'_i = \frac{1}{n} \sqrt{\sum_{j=1}^n g_i(\mu_{B_{ij}})(1 - g_i(\mu_{B_{ij}}))}. \quad (4)
\]

If the neighborhood area is sufficiently small, \( \mu_{B_{ij}} \) is close to \( \mu_{B_i} \). Then \( EX'_i \approx EX_i, \sigma X'_i \approx \frac{1}{\sqrt{n}}\sigma X_i \). Therefore the relation in (1) also holds for the neighborhood of \( X_i \) in place of \( X_i \) itself, with smaller variance. If all partitions beyond depth \( d + 1 \) is ignored, \( X'_i + d \) is essentially the average block depth among \( B_{ij} \). This suggests a more consistent similarity between the average block depth of co-located neighborhoods.

4. BLOCK STRUCTURE INFERENCE MODEL

4.1. Fast block structure determination

Multiple resolutions encoding are often done in parallel, where the same video is encoded to different resolutions on a multi-core server, and the overall time cost depends on the most time-consuming encoding process, i.e. the one with the highest target resolution. As a result, a fast block structure determination algorithm that accelerates high resolution encoding also reduces overall time cost by the same amount.

The block structure similarity among co-located neighborhoods, as proven by (1), can be exploited to provide useful information about block structures for high resolution encoding. Using the same terminology, the objective can be stated as follows. It is to be decided whether \( B_1 \) should be partitioned, which relies on the distribution of \( X_1 \), a function of \( EX_2 \). Since \( P_2 \) takes shorter time than \( P_1 \), the partition results near \( B_2 \) should be available, i.e. for each \( B_{ij} \), the observed partition result is \( \tilde{X}_{2j} \), hence the observed value of \( X'_2 \), which is the sample mean \( \tilde{X}_{2j} \), is always available.

Now, (3)(4) show that \( EX_2 \) can be approximated with the sample mean \( \tilde{X}_{2j} \) while keeping variance small. By (1), \( g_1 \circ g_2^{-1} \) also needs to be determined, however this can only be done with history statistics, in other words, the value of \( X_1 \) is needed. In a fast determination algorithm, instead of running full RDO in \( P_1 \), the encoding results from \( P_2 \) is used to make inference, which may result in suboptimal decisions and makes obtaining \( X_1 \) impossible.

Therefore, the fast determination algorithm is occasionally disabled to evaluate \( X_1 \). Coupled with \( \tilde{X}_{2j} \) which is always available, the information about \( g_1 \circ g_2^{-1} \) can be obtained by (1). Once \( g_1 \circ g_2^{-1} \) is determined, the fast determination algorithm is relatively simple: compute \( EX_1 \) with (1), if it is sufficiently small, i.e. \( P(X_1 = 0) \) is sufficiently large, the RDO is terminated so that \( B_1 \) remains non-split without traversing all its partition possibilities. In this way, the RDO is shortened and time saving is achieved. In our implementation, for every 50 frames, the first 5 frames are encoded without the fast determination algorithm. This ensures that the majority of frames (90%) are encoded with acceleration, and the statistical model is always up to date. Fig. 3 shows the procedure of our accelerated encoding system. For every 50 frames, the first 5 frames are fully encoded both in \( P_1 \) and \( P_2 \), which give observed value of \( X_1 \) and \( \tilde{X}_{2j} \), respectively.
They are used to update the inference model described in the next subsection. For the rest of the frames, the updated inference model, together with estimates of $EX_2$ from $P_2$, yield estimates of $EX_1$ for fast block structure determination.

4.2. The inference model

It remains to be shown how to choose a neighborhood with suitable size to compute $X'_2$, how to deal with $g_1 \circ g_2^{-1}$, and how the algorithm affects performance.

The algorithm may fail in two scenarios: (i) $B_1$ should not be partitioned but was given a high $EX_1$ estimate, unnecessary RDO quadtree search is conducted which increases time cost; (ii) $B_1$ should be partitioned but is given a low $EX_1$ estimate, its subsequent RDO is terminated and results in RD loss. If “$B_1$ should not be partitioned” is our null hypothesis, (i) is the type I error, and (ii) is the type II error. In practical situations, the resulting RD loss needs to be limited, while reducing the time cost as much as possible. Therefore, a type II error rate threshold $\epsilon$ is set, an algorithm that has a type II error rate smaller than $\epsilon$, and minimizes the type I error rate is ideal for our purpose. A smaller $\epsilon$ reduces the type II error, improving the RD performance, but in turn increases the type I error and reduces time savings, and vice versa. Therefore, $\epsilon$ can be used to control the trade-off between acceleration and RD performance.

According to the previous section, our criteria for terminating $B_1$’s RDO is whether the estimated $EX_1$ is small. Therefore the inference model also includes a threshold $\tau$ that decides if $EX_1$ is sufficiently small. Since $EX_2$ is used to compute $EX_1$ by (1), and $g_1 \circ g_2^{-1}$ is monotonically increasing, the criteria can instead be defined based on $EX_2$, i.e. if $EX_2 < \tau$, the predicted value of $X_1$ (denote by $X'_1$) is 0, and vice versa. The type I error occurs when $X'_1 = 1$ and the observed value $X_1 = 0$, and the type II error occurs when $X'_1 = 0$ and $X_1 = 1$.

The type I/II errors come from two factors, the randomness of $X_1$, and the inaccuracy in estimating $EX_2$. It is thus important to choose a suitable sized neighborhood to provide accurate and consistent estimation of $EX_2$. To this end, a few more assumptions are needed to analyze (5). Fix $B_2$, denote the vector from $B_2$ to $B_{2j}$ by $\eta_j$, assume $\mu B_{2j}$ to be continuous as $B_{2j}$ moves on the frame plane, it is natural to model it with a generalized two dimensional Wiener process:

$$
\mu B_{2j} \sim \mathcal{N}\left(\mu B_2 + \beta \cdot \eta_j, \|\eta_j\| \sigma^2\right),
$$

where vector $\beta$ represents the deterministic drift, and $\sigma^2$ represents the uncertainty. Furthermore assume that $g_2$ is differentiable and has a first order Taylor expansion at $\mu B_2$:

$$
g_2(\mu B_{2j}) = g_2(\mu B_2) + g'_2(\mu B_2)(\mu B_{2j} - \mu B_2),
$$

by (4), for the sample mean $X'_2$, which is our estimator of $EX_2$, its bias satisfies

$$
|EX'_2 - EX_2| = \frac{1}{n} |g'_2(\mu B_2)\beta \sum \eta_j|
\leq \frac{1}{n} |g'_2(\mu B_2)| \|\beta\| \sum \|\eta_j\|.
$$

From (4), there is a bias-variance trade-off in choosing the size of the neighborhood. For if a large neighborhood of $B_2$ is chosen, the number of neighbor blocks $n$ that have the same size as $B_2$ increases, which gives a smaller variance due to (4). However, a large neighborhood also cause $\|\eta_j\|$ to be large, resulting in a large bias upper bound by (7). It is also noteworthy that by (5), a large $\|\eta_j\|$ causes the variance of $X_{2j}$ to increase, offsetting (4). To sum up, there is a best neighborhood size that balances the bias and variance of the estimator $X'_2$, neither too large nor too small.

The discussion above assumed that a large neighborhood always leads to a large $n$ in (4). This is not always the case, especially when the depth of $B_2$ is large, as most of its neighbor blocks remain non-split at lower depths. Also, when $n$ is small, the spatial distribution of $B_{2j}$ is less likely to be balanced around $B_2$, in the sense that the vector sum $\sum \eta_j$ in (7) is less likely to cancel out. In this case, a large neighborhood significantly increases the bias, and fails to reduce the variance. As a result, for small sized blocks, $X'_2$ itself rather than $X'_2$ is used to estimate $EX_2$ (equivalently, the neighborhood size is set to zero). In fact, small blocks are so unlikely to be partitioned, that the number of type I errors often vastly exceeds the number of type II errors. In this case, the neighborhood size is chosen to minimize the total number of type I and II errors, without restricting the type II error rate.

4.3. Implementation

In our implementation, the neighborhood of a square block is a square formed by expanding the block’s edge in each direction by a specified margin, see Fig. 4. Based on our previous analysis, the implementation of our fast block structure determination algorithm is described as follows.
For every 50 input frames, the first 5 frames are both fully encoded in \( P_1 \) and \( P_2 \).

- For each \( d \in \{0, 1, 2\} \), all square blocks with depth \( d \) (regardless of further partition) is found in \( P_1(P_2) \), denote them by \( \Omega_1(\Omega_2) \). For any block \( B_1 \) in \( P_1 \), \( X_1 \) is the partition choice of \( B_1 \), and \( \bar{X}_2 \) is the average block depth of \( \Omega_2 \) within the co-located neighborhood, ignoring further partitions beyond depth \( d + 1 \). The best margin and the corresponding \( \tau \) are searched from a discrete set of values. The threshold \( \tau \) is chosen as the largest value as long as the type II error \((\tilde{X}_2 < \tau, \tilde{X}_1 = 1)\) rate does not exceed the error rate threshold \(\epsilon\), in this way the type I error \((\tilde{X}_2 \geq \tau, \tilde{X}_1 = 0)\) rate is automatically minimized. The best margin is then the one that gives the smallest type I error rate.

- For \( d = 3 \), the margin is set to 0, and the threshold \( \tau \) is chosen such that the total number of type I and II errors is minimized.

The best margin and the corresponding threshold \( \tau \) are recorded for each depth. In our implementation, margins are chosen from multiples of 8 in [8, 128], and \( \tau \) is chosen from multiples of 0.1 in [0, 1]. This offers sufficient granularity with small searching complexity. If the neighborhood only partially covers a block, the percentage of the covered area is used as weight to compute the average depth.

The rest 45 frames are fully encoded in \( P_2 \). For a block \( B_1 \) in \( P_1 \), \( \bar{X}_2 \) is computed from the encoding results of \( P_2 \), using the best margin of the corresponding depth. Then \( \bar{X}_2 \) is compared to the corresponding \( \tau \) of the margin. if \( \bar{X}_2 < \tau \), \( B_1 \) remains unpartitioned. In other words, RDO only considers non-split and 2-way splits. Otherwise, the ordinary RDO is conducted.

### 5. EXPERIMENTAL RESULTS

To demonstrate the effectiveness of our inference model, the fast block structure determination algorithm is integrated into the AV1 encoder to encode 8 test sequences from CTC [13], with native resolutions of 1280×720 and 1920×1080 (see Table[1]), each consists of 150 frames, using a constant QP of 22, 27, 32, 37, the key frame interval is set to 50. For 720p sequences, they are encoded to target resolutions of 1280×720 and 960×540. For 1080p sequences, they are encoded to target resolutions of 1920×1080 and 1440×810.

The high target resolution encoding process is accelerated using our proposed algorithm. The original AV1 encoder is then used to encode the same sequences with the same high target resolutions. The RD performance and time cost are compared, using BD-rate [14], BD-PSNR [15], and the total time cost of high resolution encoding processes of all QP’s. Only time costs of high resolution encoding processes are compared, since, as stated before, the overall time cost is equal to that of the high resolution encoding process, if multiple resolutions encoding are done in parallel.

Two sets of experiments are conducted, where the type II error rate threshold \( \epsilon \) is set to 0.1 (Table[1]) and 0.2 (Table[2]), respectively, to demonstrate its capability in controlling the trade-off between acceleration and RD performance. The column \( \Delta T \) is the time reduction of our algorithm comparing to

### Table 1. Encoding results for \( \epsilon = 0.1 \)

| Sequence               | BD-rate | BD-PSNR | \( \Delta T \) |
|------------------------|---------|---------|---------------|
| FourPeople (720p)      | 0.51%   | -0.007dB| -33.2%        |
| Johnny (720p)          | 0.90%   | -0.010dB| -35.6%        |
| Kristen&Sara (720p)    | 0.73%   | -0.010dB| -29.2%        |
| BasketballDrive (1080p)| 0.58%   | -0.008dB| -32.5%        |
| Cactus (1080p)         | 0.85%   | -0.013dB| -28.6%        |
| Kimono (1080p)         | 0.73%   | -0.018dB| -26.9%        |
| ParkScene (1080p)      | 0.73%   | -0.018dB| -31.7%        |
| **Average**            | 0.71%   | -0.017dB| -30.1%        |

### Table 2. Encoding results for \( \epsilon = 0.2 \)

| Sequence               | BD-rate | BD-PSNR | \( \Delta T \) |
|------------------------|---------|---------|---------------|
| FourPeople (720p)      | 0.89%   | -0.014dB| -43.7%        |
| Johnny (720p)          | 0.95%   | -0.010dB| -45.1%        |
| Kristen&Sara (720p)    | 1.14%   | -0.015dB| -39.5%        |
| BasketballDrive (1080p)| 0.66%   | -0.010dB| -34.7%        |
| Cactus (1080p)         | 0.95%   | -0.017dB| -31.8%        |
| Kimono (1080p)         | 0.91%   | -0.021dB| -29.7%        |
| ParkScene (1080p)      | 0.84%   | -0.020dB| -34.8%        |
| **Average**            | 1.04%   | -0.030dB| -36.8%        |

### Table 3. Encoding results for algorithm in [12]

| Sequence               | BD-rate | BD-PSNR | \( \Delta T \) |
|------------------------|---------|---------|---------------|
| FourPeople (720p)      | 2.80%   | -0.045dB| -42.8%        |
| Johnny (720p)          | 2.00%   | -0.023dB| -46.4%        |
| Kristen&Sara (720p)    | 2.03%   | -0.029dB| -42.2%        |
| BasketballDrive (1080p)| 3.37%   | -0.230dB| -31.5%        |
| Cactus (1080p)         | 1.78%   | -0.023dB| -30.7%        |
| Kimono (1080p)         | 1.19%   | -0.020dB| -29.7%        |
| ParkScene (1080p)      | 1.95%   | -0.027dB| -31.6%        |
| **Average**            | 1.82%   | -0.048dB| -23.2%        |
the original AV1 encoder. It is observed that $\epsilon = 0.1$ achieves 30.1% average time reduction with a negligible 0.71% BD-rate (0.017dB BD-PSNR loss), while $\epsilon = 0.2$ achieves a higher average time reduction of 36.8% but also a higher 1.04% BD-rate (0.03dB BD-PSNR loss).

For comparison, the latest multiple resolutions encoding algorithm [12] from literature is also evaluated using the same test settings. Although [12] is based on HEVC and best suited for low resolution encoding acceleration, it can be migrated to AV1 without difficulty and accelerate high resolution encoding. Table 3 shows the performance of this algorithm. Comparing to our proposed algorithm where $\epsilon$ is set to 0.2, they both achieve about 35% average time reduction, but the BD-rate (BD-PSNR loss) of our proposed algorithm is halved. This proves the effectiveness of our proposed algorithm in dealing with high resolution encoding acceleration.

6. CONCLUSIONS

In this paper, we consider the problem of encoding the same video to different target resolutions using AV1. We first present the block structure similarity across different resolutions, and a model based on fineness of frame detail and scale of object motions is proposed to analyze the similarity. We later see that this model can be used to derive an inference model that accelerates high resolution encoding based on low resolution encoding results. The average block depth of the co-located neighborhood is used to decide early termination in the RDO process. A bias-variance trade-off can be achieved by searching for an optimal neighborhood range. Experimental results show that our proposed algorithm offers the capability to control the trade-off between RD performance and time reduction, achieving 30.1%-36.8% time reduction while keeping BD-rate low at 0.71%-1.04%.

7. REFERENCES

[1] O. Oyman and S. Singh, “Quality of experience for http adaptive streaming services,” IEEE Communications Magazine, vol. 50, no. 4, pp. 20–27, April 2012.
[2] Bo Li and Jiangchuan Liu, “Multirate video multicast over the internet: an overview,” IEEE Network, vol. 17, no. 1, pp. 24–29, Jan 2003.
[3] AOM, “Av1 codec library,” https://aomedia.googlesource.com/aom/
[4] D. Schroeder, P. Rehm, and E. Steinbach, “Block structure reuse for multi-rate high efficiency video coding,” in 2015 IEEE International Conference on Image Processing (ICIP), Sept 2015, pp. 3972–3976.
[5] D. Schroeder, A. Ilangovan, M. Reisslein, and E. Steinbach, “Efficient multi-rate video encoding for hevc-based adaptive http streaming,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 25, no. 8, pp. 1423–1434, Aug 2015.
[6] J. De Praeter, A. J. Daz-Honrubia, N. Van Kets, G. Van Wallendael, J. De Cock, P. Lambert, and R. Van de Walle, “Fast simultaneous video encoder for adaptive streaming,” in 2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP), Oct 2015, pp. 1–6.
[7] L. Ahmad, Xiaohui Wei, Yu Sun, and Ya-Qin Zhang, “Video transcoding: an overview of various techniques and research issues,” IEEE Transactions on Multimedia, vol. 7, no. 5, pp. 793–804, Oct 2005.
[8] Y. Chen, Z. Wen, J. Wen, M. Tang, and P. Tao, “Efficient software h.264/avc to hevc transcoding on distributed multicore processors,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 25, no. 8, pp. 1423–1434, Aug 2015.
[9] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable video coding extension of the h.264/avc standard,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 9, pp. 1103–1120, Sept 2007.
[10] A. Zaccarin and Boon-Lock Yeo, “Multi-rate encoding of a video sequence in the dct domain,” in 2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No.02CH37353), 2002, vol. 2, pp. II–680–II–683 vol.2.
[11] D. H. Finstad, H. K. Stensland, H. Espeland, and P. Halvorsen, “Improved multi-rate video encoding,” in 2011 IEEE International Symposium on Multimedia, Dec 2011, pp. 293–300.
[12] D. Schroeder, A. Ilangovan, and E. Steinbach, “Multi-rate encoding for hevc-based adaptive http streaming with multiple resolutions,” in 2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP), Oct 2015, pp. 1–6.
[13] F. Bossen, “Common test conditions and software reference configurations,” JCT-VC, Tech. Rep. H1100, 2012.
[14] G. Bjontegaard, “Improvements of the bd-psnr model,” in ITU-T SG16 Q. 2008, vol. 6, p. 35.
[15] G. Bjontegaard, “Calculation of average psnr differences between rd curves,” in Doc. VCEG-M33 ITU-T Q6/16, 2001.