Adaptive Bitrate Selection for Video Encoding with Reduced Block Artifacts

Min-Kook Choi, Hyun-Gyu Lee, Minseok Song, Sang-Chul Lee*
Department of Computer Engineering, Inha University, Incheon, Korea
{mkchoi,hg_lee}@inha.edu, {mssong,sclee}@inha.ac.kr

ABSTRACT

Blocking artifacts, commonly introduced during video encoding, are one of the major causes of reduced perceptual video quality. The trade-off between these artifacts and bitrate can be improved by adaptively selecting frames from a set of video copies encoded at different bitrates, prior to actual video encoding. We propose a new direction of constructing mixed bitrate video based on content-based image analysis on each video frame, which was posed as a problem of pre-analysis for the final video encoding step.

The proposed method consists of the following steps: First, we define a simple and fast impact metric in order to identify the blocking artifacts in each frame of multiple videos, encoded at different bitrates. Based on the impact metric, we generate a blocking artifact density functions for the available bitrates, on the whole video. Finally, we define and optimize our objective function from the blocking artifact density functions in order to select a bitrate with minimum perceptual blockiness and file size for each frame.

We validated our method throughout multiple types of videos, showing improved visual quality for the same file size based on commonly used quality assessment measures, such as MSU blocking, MSU blurring, SSIM, 3SSIM, and stSSIM. The reduction rates of average file size and average blocking artifact were about 4.9% and 8.3% over maximum bitrate encoding, respectively.

Keywords

Adaptive bitrate selection; video encoding; video quality measurement (VQM); blocking artifact

1. INTRODUCTION

Encoding is an essential process in the provision of video streaming services, and there has been a variety of approaches to improve efficiency and visual quality. Many new encoding tools and content-based metrics have been introduced to improve the compactness of video coding standards. Recently, the High Efficiency Video Coding (HEVC) [14] shows a significant improvement in compression over previous standards [12]. Another approach to improve the compactness of video coding is adaptive bitrate control, which also facilitates dynamic power control in embedded systems, network bandwidth adjustment, and video quality optimization. Initial work on adaptive bitrate control was mostly concerned with the support of network bandwidth renegotiation [13, 17]. For example, Javdtalab et al. [6] developed an algorithm that measures network bandwidth, frame size, PSNR and SSIM for rate-control under H.264 HEVC. Adaptive bitrate control has recently escaped from the network environment, and been used to improve the efficiency of video encoding and transcoding, and to reduce the power consumption of devices. Feghali et al. [4] proposed the variation of frame-rate and quantization parameters to achieve adaptive bitrate control in mobile video broadcasting, subject to a perceived-quality metric.

Closely related to our research, another lines of method to enhance encoding efficiency have been proposed to re-

*Corresponding author (email: sclee@inha.ac.kr)

MM '16, October 15-19, 2016, Amsterdam, Netherlands

DOI: http://dx.doi.org/10.1145/2964284.2967227

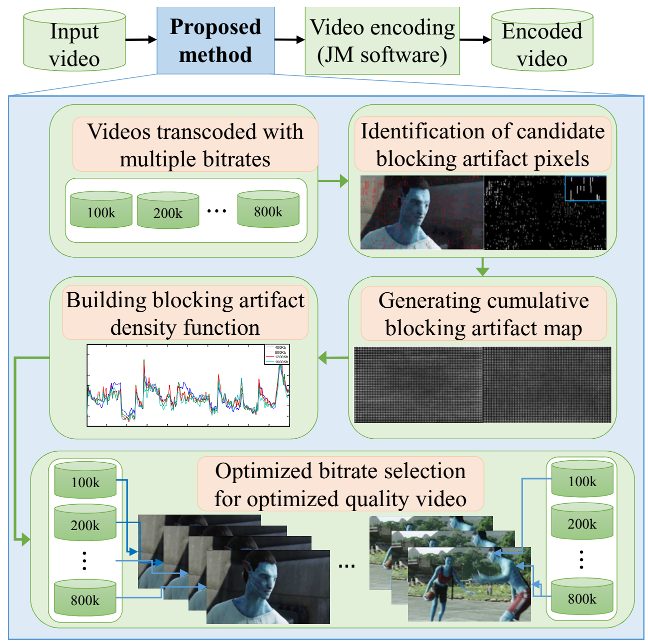

Figure 1: Overview of the proposed adaptive bitrate selection method.

duce blocking artifact. Li et al. presented blocking artifact measurement metric with no reference information using Tchebichef moments [9]. They used an observation of multiple orders of the Tchebichef kernels which have varying abilities to capture blockiness. On the other hands, Yoo et al. developed discrete cosine transform (DCT) based interblock correlation for block artifact reduction algorithm[18].

We introduce a new approach of video quality optimization prior to actual video encoding. In specific, we propose a framework of constructing a multi-bitrate video based on content-based image analysis on each video frame, which is posed as a problem of pre-processing step before the actual video encoding step, as summarized in Figure 1. The proposed method has the following main differences from a multi-pass VBR: VBR is based on the motion changes, which does not necessarily reflect blocking artifacts, but the proposed method uses the pre-computed measures of blocking artifacts to select the video frames with optimal bitrates from the decoded videos that have been already encoded in different bitrates. Therefore, our method can measure/predict them before encoding the final video, whereas VBR cannot measure structural similarity before actually encoding the final video. It means that we can fine-tune the video to generate one with allowable levels of blocking artifacts associated with structural similarity, with budgeting file size. Since the proposed technique is an independent process of typical video quality enhancement techniques, that are typically applied after decoding process, our technique can still be applied as a standalone or along with other post processing techniques such as [7].

Although many different techniques can be applied for frame quality analysis problem, generally, blocking artifacts are known to be caused by quantization errors at the boundary location between adjacent macroblocks where there is a large difference in quantization levels, and to have a serious impact on the visual quality of videos [10]. First, we identify the blocking artifacts in each frame of the video, encoded at different bitrates, and quantify them using an impact metric. Then, we generate an artifact map for the whole video, consisting of blocking artifact density functions for the available bitrates. Finally, we construct an objective function from the blocking artifact density functions, and select a bitrate for each frame by L_1 and L_2 optimization of this objective function. Since we need to encode an input video into multiple bitrates before reconstructing a multi-bitrate video, note that the proposed method is intended for archived video contents, e.g., video-on-demand service, rather than real-time encoding, e.g., web casting. However, the proposed method is practical in real world application since once the videos in multiple bitrates are created, which is the most time consuming process, frame selection and reconstruction can be processed relatively in small amount of time. Furthermore, differently reconstructed videos can be generated in nominal amount of time.

The main contribution of the proposed method is twofold: First, we explored a new concept of optimizing visual video quality as a pre-processing step. The proposed method was validated throughout multiple types of videos in our experiments, showing improved visual quality for the same file size based on commonly used assessment measures, such as MSU blocking, MSU blurring, SSIM, 3SSIM, and stSSIM. Second, we introduced an optimization method with blocking artifact density function by incorporating a fast and effective blocking artifact detection method. By adjusting parameters, it is also possible to adjust detecting sensitivities according to different displaying environments. We tested multi-rate encoding on a well- known video dataset for testing MPEG-4 and mobile high-definition (HD) video. Our encoding method shows promising reduction rates in file size and decoding time, while improving visual quality when measured by standard video quality metrics (VQMs) .

2. BLOCKING ARTIFACTS

We search for blocking artifacts in multiple encodings of an input videos. After encoding at different bitrates, blocking artifacts are identified by examining the horizontal and vertical variations in one-dimensional derivative of pixel intensity in each frame. A cumulative map is then constructed from these candidate pixels, which is then used to locate the blocking artifacts in each frame. We will use $\mathbf{V} = (f^{(1)}, f^{(2)}, ..., f^{(k)})$ to denote an input video with k frames. By encoding $\mathbf{V}$ at h different bitrates, we obtain a set of h videos $\mathcal{V} = \{\mathbf{V}_1, \mathbf{V}_2, ..., \mathbf{V}_h\}$.

Artifacts Identification. Blocking artifacts typically occur where are large differences in quantization level between neighboring macroblocks, and the artifacts appear at the edges of each macroblock in the encoded frame. We calculate the one-dimensional derivatives of pixel intensity, in both horizontal and vertical directions $(f_x^{(i)}, f_y^{(i)})$, in the i^{th} frame $f^{(i)} \in \mathbb{N}^{m \times n}$ of a video where m and n are the width and height of a frame. The frame $f^{(i)}$ belongs to the j^{th} encoded video $\mathbf{V}_j$, and the derivatives $(f_x^{(i)}, f_y^{(i)})$ are defined as $f_x^{(i)}(u,v) = f^{(i)}(u,v) - f^{(i)}(u+1,v)$ and $f_y^{(i)}(u,v) = f^{(i)}(u,v) - f^{(i)}(u,v+1)$, where $f^{(i)}(u,v)$ is the intensity of the pixel at location (u,v) in the i^{th} video frame.

A one-dimensional horizontal vector $\mathbf{v}_x(u,v) = [\Delta_x^1, \Delta_x^2, ..., \Delta_x^s]^T$ is constructed for the pixel at (u,v), and $\mathbf{v}_x(u,v)$ is the variation in intensity from $f_x^{(i)}(u,v)$ to $f_x^{(i)}(u,v+s)$, where s is the number of consecutive pixels with the same intensity which form the boundary of a visible block that has been detected. A second vector $\mathbf{v}_y$ is constructed in the vertical direction from $f_y^{(i)}(u,v)$ to $f_y^{(i)}(u+s,v)$. This process can be regarded as mapping $f_x^{(i)} \rightarrow \mathbf{W}_x^{(i)}$, where $\mathbf{W}_x^{(i)} \in \mathbb{N}^{m \times n \times s}$ is a 3D array. A matrix $\mathbf{D}^{(i)}$ can then be defined given as a constraint on $\mathbf{W}_x^{(i)}(u,v)$: $\mathbf{D}_x^{(i)}(u,v) = \begin{cases} 1, & \text{if } \Delta_x^1 = \Delta_x^2 = ... = \Delta_x^s \\ 0, & \text{otherwise} \end{cases}$, such that, $\mathbf{v}_x(u,v) \neq \mathbf{0}$. $\mathbf{D}^{(i)}(u,v) = \mathbf{D}_x^{(i)}(u,v) \vee \mathbf{D}_y^{(i)}(u,v)$ is a binary matrix of the same size as $f^{(i)}$, which indicates the likelihood of a pixel contributing to a blocking artifact.

Blocking Artifact Map. $\mathbf{C}_j$ is the blocking artifact map for the j^{th} encoded video $\mathbf{V}_j$ in $\mathcal{V}$. The map $\mathbf{C}_j$ is constructed from $\mathcal{D}_j = \{\mathbf{D}_j^{(1)}, \mathbf{D}_j^{(2)}, ..., \mathbf{D}_j^{(k)}\}$ for all the pixel locations (u,v) on all the video frames, where $\mathcal{D}_j$ is a set of matrices of artifact pixels and k is the number of frames in $\mathbf{V}_j$: $\mathbf{C}_j(u,v) = \sum_{i=1}^{k} \mathbf{D}_j^{(i)}(u,v)$. We normalize this map to $\tilde{\mathbf{C}}_j = \mathbf{C}_j / \parallel \mathbf{C}_j \parallel$. By processing all the videos in $\mathcal{V}$, we obtain a set of blocking artifact maps $\mathcal{C} = \{\tilde{\mathbf{C}}_1, \tilde{\mathbf{C}}_2, ..., \tilde{\mathbf{C}}_h\}$, where the j^{th} element in $\mathcal{C}$ corresponds to the encoded video $\mathbf{V}_j$. If the value at pixel location (u,v) in a normalized blocking artifact map $\tilde{\mathbf{C}}_j$ is greater than a threshold ρ then

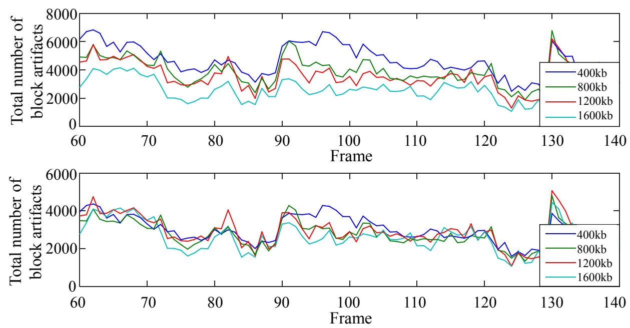

Figure 2: (a) An example blocking artifact density functions for 80 frames of a video encoded at four different bitrates ($s = 8$). (b) Scaled blocking artifact density functions $\mathcal{L}'$ ($h = 4$, $p = 2$). The block artifact functions with different bitrate are scaled by λ.

a visible blocking artifact can appear at (u, v). We consider $\tilde{\mathbf{C}}_j(u, v)$ to be the candidate location of a blocking artifact if it satisfies the constraint: $\mathbf{R}_j(u, v) = \begin{cases} 1, & \text{if } \tilde{\mathbf{C}}_j > \rho \\ 0, & \text{otherwise} \end{cases}$. We call $\mathbf{R}_j \in \mathbb{N}^{m\times n}$ the region matrices corresponding to $\mathbf{V}_j$: we construct a set of region matrices $\mathcal{R} = \{\mathbf{R}_1, ..., \mathbf{R}_h\}$. We now obtain $\mathbf{A}^{(i)} \in \mathbb{N}^{m\times n}$ corresponding to the i^{th} input video frame $f^{(i)}$ in $\mathbf{V}_j$ by intersecting $\mathbf{D}_j^{(i)}$ with the region matrix $\mathbf{R}_j$: $\mathbf{A}_j^{(i)}(u, v) = \mathbf{D}_j^{(i)}(u, v) \wedge \mathbf{R}_j(u, v)$. The set $\mathcal{A} = \{\mathbf{A}_1, ..., \mathbf{A}_h\}$ is later used to generate blocking artifact density functions, allowing us to select frames from $\mathcal{V}$ with the required visual quality.

3. BITRATE SELECTION

We now show how to generate blocking artifact density functions using the set of blocking artifact matrices $\mathcal{A}$, and how to choose the best bitrate for each frame optimizing an objective function.

Blocking Artifact Density Function. The outcome l_i is the visual impact of all the blocking artifacts occurring in the i^{th} frame $f^{(i)}$, and can be expressed as $l_i = \sum_{u=1}^{m} \sum_{v=1}^{n} \mathbf{A}^{(i)}(u, v)$. After repeatedly applying this process, we assemble a density function $\mathbf{l}_j = [l_{1j}, l_{2j}, ..., l_{kj}]^T$, where k is the number of frames in $\mathbf{V}_j$. We also obtain a set of h vectors $\mathcal{L} = \{\mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{l}_h\}$ from the set $\mathcal{V}$, and then generate h blocking artifact density functions. Figure 2 (a) shows an example of these functions when a video is encoded at four different bitrates ($h = 4$).

Objective Function Optimization. To select the best bitrate in each frame for an input video, we define an objective function that expresses the difference between the blocking artifact density functions at the lower bitrates and the same function at the highest bitrate:

$$\underset{\Lambda}{\operatorname{argmin}} \sum_{j=1}^{h-1} \parallel \mathbf{l}_h - \lambda_j \mathbf{l}_j \parallel_p . \quad (1)$$

where $\mathbf{l}_h$ is the blocking artifact density function for the highest bitrate in the set of density functions $\mathcal{L}$, and λ_j is a weight applied to the j^{th} density function. We minimize this function using L_p distance to obtain a vector of weight coefficients $\Lambda = [\lambda_1, \lambda_2, ..., \lambda_{h-1}]^T$. The exponent of the p-norm is set to 1 or 2 in our experiments. Minimization produces a set $\mathcal{L}'$ of scaled density functions, as follows:

$$\mathcal{L}' = \{\mathbf{l}_1', \mathbf{l}_2', ..., \mathbf{l}_{h-1}'\}, \quad \mathbf{l}_j' = \lambda_j \mathbf{l}_j, \quad (2)$$

where the j^{th} density function $\mathbf{l}_j'$ is the scaled density function from $\mathbf{l}_j$ by λ_j, and $1 \le j \le h-1$. Figure 2 (b) shows an example of $\mathcal{L}'$ with $\mathbf{l}_h$ when h is 4 and p is 2. Since $\mathbf{l}_j'$ is a column vector with k elements, we obtain a $k \times h$ matrix $Q \in \mathbb{N}^{k\times h}$ from $\mathcal{L}'$ as follows:

$$\mathbf{Q} = \begin{bmatrix} l_{11}' & l_{12}' & \cdots & l_{1h-1}' & l_{1h}' \\ l_{21}' & l_{22}' & \cdots & l_{2h-1}' & l_{2h}' \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ l_{k1}' & l_{k2}' & \cdots & l_{kh-1}' & l_{kh}' \end{bmatrix}. \quad (3)$$

Now, we select the best bitrates from the matrix $\mathbf{Q}$ as follows:

$$\hat{\mathbf{l}} = [\hat{l}_1, \hat{l}_2, ..., \hat{l}_k]^T, \quad \hat{l}_i = \min(l_{i1}', l_{i2}', \cdots, l_{ih}') \quad (4)$$

where the elements of $\hat{\mathbf{l}}$ are the selected bitrates for all the frames in a video. Finally, we apply a one-dimensional median filter to $\hat{\mathbf{l}}$ in order to prevent sudden changes of bitrate between two consecutive frames.

Video Generation with Variable Bitrates. Using the selected frame information vector $\hat{\mathbf{l}}$, we generate a new video $\hat{\mathbf{V}}$ from the h videos in $\mathcal{V}$. The new video $\hat{\mathbf{V}}$ has k frames, just like the original video $\mathbf{V}$: each frame $\hat{f}$ in $\hat{\mathbf{V}}$ is selected from the h available encoded frames based on the information vector $\hat{\mathbf{l}}$. The quality of the encoded video $\hat{\mathbf{V}}$ depends on the minimum and maximum bitrates of the encoded videos in $\mathcal{V}$ and the number of videos h with different bitrates in that range. The results of varying these two parameters are described in next section. Finally, the final video can be generated using a standard encoding technique either one-pass or two-pass VBR encoding approach.

4. EXPERIMENTAL RESULTS

To the best of our knowledge, there exists no other video pre-processing technique that can be directly compared with the proposed technique. The closest application may be MPEG-DASH (Dynamic Adaptive Streaming over HTTP), mainly used by Netflix, but it is still uncomparable since they change the bitrate depending on the network bandwidth, not to maintain the QoE. Our scheme aims at minimizing blocking artifact subject to the file size, which allows users in the DASH environment to receive higher-quality video segments, compared with simple VBR transcoding under the same network condition.

In our experiments, we compared our technique with the test videos encoded in multiple bitrates with H.264/AVC codec as baselines. We show the experimental results from our proposed technique applied to MPEG-4 [3] and mobile HD test datasets. The MPEG-4 dataset contains several types of video clip where each consists of frames at 352×288 standard definition (SD) resolution, with a frame-rate of 60fps. The mobile HD test dataset contains six videos with about 30 shots; their spatial resolution is 800×480, and there are $9,000$ frames at 30fps. This dataset is included to determine whether our method is applicable to rather

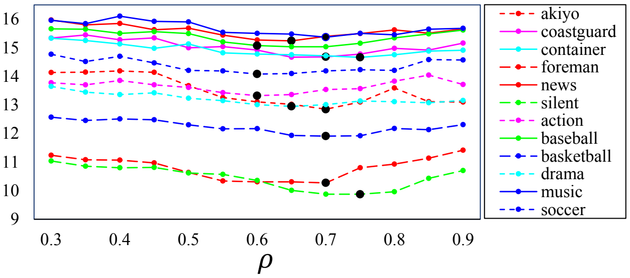

Figure 3: Parameter selection for the threshold ρ with all videos. Black dot indicates lowest MSU blocking values in each video.

long and high-resolution sequences. All the videos in both dataset are encoded using the standard H.264/AVC codec with the H.264/AVC JM reference software [15][2] after applying our preprocessing technique.

Parameter Selection. We set the number of bitrate levels h to 8, with the same interval between each bitrate. We created the h transcoded videos in $\mathcal{V}$ using the FFmpeg encoder [1]. The minimum bitrates for the videos in the MPEG-4 test dataset were selected between $40kb/s$ to $275kb/s$, and the maximum bitrates between $110kb/s$ to $625kb/s$, depending on the amount of data in the video. We encoded these mobile HD dataset videos at bitrates between $900kb/s$ and $1600kb/s$. The calculation of one-dimensional variation in intensity is controlled by the length s of the vectors $\mathbf{v}_x$ and $\mathbf{v}_y$ used to detect artifacts, and the distance metric p in the optimization. We tuned the parameters s and p to account for the average of blocky noise and file sizes, using the three videos in the mobile HD dataset with $h = 8$. As a result of this tuning exercise, we selected s to 8 and p to 2. Threshold ρ is a parameter to impose constraint to the normalized blocking artifact map in $\mathcal{C}$. We selected $\rho = 0.6791$ by considering average reduction amount of blocky noises in every video as shown in Figure 3.

Video Quality Measurement Results. We used five video quality metrics (VQMs) provided by the MSU quality measurement [5] for quantitative assessment of our technique: structural similarity (SSIM) [16], 3-component structural similarity (3SSIM) [8], spatio-temporal structural similarity (stSSIM) [11], MSU blurring, and MSU blocking [5]. The structural similarity metric compares the videos with a reference video (in this case an unencoded video) in terms of luminance, contrast and the structural properties of each frame. The 3SSIM augments the results from SSIM with edge, texture, and smoothness information; and stSSIM provides temporal information. All these similarity metrics have a value range of 0 to 1.0, with higher values corresponding to better perceptual quality. MSU blurring estimates sharpness, and again higher values are better. MSU blocking metric is particularly significant for our experiments, because our aim is to reduce the numbers of blocking artifacts. Higher values of this metric correspond to higher levels of blocking artifact noise.

Our result shows that the proposed method generated a video with comparable file size to the video encoded in the minimum bitrate, while achieving less or equal blocky noise than the video encoded the maximum bitrate. The structural similarity values are also comparable to the videos with maximum bitrate with meaningful reduction in file size. Table 1 shows the average file sizes and the VQMs for all the MPEG-4 test and mobile HD datasets, and Table 2 shows the average file size and blocking artifact reduction results depending on different categories and types of video clips. Proposed method achieves about 4.9% and 8.7% reduction in the average file size and the blocking artifacts in all video clips, respectively. It is also observable that the reduction rates are especially increased for videos with large movements in a video. The examples of visual results are available in the supplementary material.

Reduction rate(%)	MPEG-4	Mobile HD	Active	Inactive	Avg.
File size	5.103	4.752	4.711	5.144	4.927
MSU blocking	5.266	11.277	11.985	4.559	8.271

Table 1: The average file size and blocking artifact reduction rates in different categories of video clips. For example, the best reduction in blocking artifact was achieved in "active" video clips.

5. CONCLUSIONS

We proposed a novel video pre-processing framework for efficient video encoding with improved visual quality and reduced file size through the analysis of visible blocking artifacts. As well as reduced blocky noise, processed videos are compact and achieved good measures in multiple VQMs. Especially, the proposed method is practical for off-line video service, e.g., video-on-demand, since multiple versions of video reconstruction can be created relatively in small amount of time, once the videos in multiple bitrates are created. In order to apply the proposed method, multiple-encoded videos should be prepared by off-line processing. In spite of this prerequisite, our method can be used for many lines of real world applications on video encoding frameworks, such as video on demand and video achieving.

Since our technique does not directly modify any video encoding scheme, it is applicable to any recent encoding codecs such as H.265/MPEG-H HEVC. Other analysis metrics and optimization schemes might potentially improve our results, in terms of the estimation of visual quality and computational efficiency. It may also be possible to extract features reflecting different aspects of visual quality from the spatial and temporal domains of an encoded video in future work.

6. ACACKNOWLEDGMENT

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (No. 2012M3 C4A7032781) and Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2015R1D1A1A01057968).

7. REFERENCES

[1] Ffmpeg. https://ffmpeg.org/.
[2] H.264/avc jm reference software. http://iphome.hhi.de/suehring/tml/download/.
[3] Xiph.org video test media. https://media.xiph.org/video/derf/.

Akiyo (kb/s)	File size (bytes)	MSU Blocking	MSU blurring	SSIM	3SSIM	stSSIM
40	43,361	14.078	9.2623	0.9776	0.9711	0.7697
50	45,340	14.025	9.3342	0.9833	0.9803	0.8280
60	46,648	13.936	9.4557	0.9868	0.9859	0.8645
70	47,801	14.112	9.4763	0.9888	0.9890	0.8882
80	48,430	14.181	9.5196	0.9907	0.9916	0.9078
90	48,826	13.996	9.5557	0.9919	0.9932	0.9218
100	48,898	13.880	9.5762	0.9928	0.9942	0.9324
110	49,131	13.656	9.6113	0.9937	0.9952	0.9409
Proposed	46,668	12.941	9.5110	0.9850	0.9870	0.8646
Coastguard (kb/s)	File size (bytes)	MSU Blocking	MSU blurring	SSIM	3SSIM	stSSIM
275	425,407	16.338	24.849	0.9319	0.9508	0.5248
325	435,965	15.382	25.152	0.9436	0.9601	0.5868
375	443,989	15.371	25.417	0.9525	0.9669	0.6376
425	450,475	15.129	25.614	0.9596	0.9724	0.6820
475	455,520	14.927	25.770	0.9658	0.9773	0.7213
525	457,271	14.986	25.871	0.9708	0.9807	0.7568
575	458,036	14.689	25.942	0.9747	0.9836	0.7861
625	459,262	14.643	26.020	0.9779	0.9858	0.8114
Proposed	426,723	14.671	25.293	0.9639	0.9771	0.7384
Container (kb/s)	File size (bytes)	MSU Blocking	MSU blurring	SSIM	3SSIM	stSSIM
50	72,731	16.468	20.042	0.9362	0.9408	0.4751
75	79,566	18.152	20.667	0.9567	0.9665	0.6138
100	86,080	16.052	21.144	0.9686	0.9790	0.7119
125	91,046	15.910	21.328	0.9765	0.9858	0.7792
150	93,809	15.827	21.456	0.9817	0.9898	0.8297
175	95,620	15.091	21.560	0.9852	0.9922	0.8654
200	96,704	15.520	21.673	0.9881	0.9941	0.8960
225	97,727	15.094	21.789	0.9901	0.9954	0.9156
Proposed	89,042	14.667	20.452	0.9627	0.9702	0.6699
Foreman (kb/s)	File size (bytes)	MSU Blocking	MSU blurring	SSIM	3SSIM	stSSIM
200	259,070	11.693	15.613	0.9605	0.9678	0.7130
225	263,581	11.508	15.787	0.9661	0.9728	0.7527
250	264,158	11.390	15.916	0.9706	0.9772	0.7858
275	266,008	11.354	16.005	0.9738	0.9803	0.8090
300	266,379	11.234	16.112	0.9770	0.9833	0.8332
325	266,445	11.156	16.193	0.9797	0.9859	0.8533
350	267,037	11.019	16.254	0.9816	0.9876	0.8680
375	267,453	10.992	16.317	0.9837	0.9894	0.8836
Proposed	261,235	10.274	16.242	0.9779	0.9852	0.8546
News (kb/s)	File size (bytes)	MSU Blocking	MSU blurring	SSIM	3SSIM	stSSIM
110	135,748	16.199	17.843	0.9792	0.9808	0.7651
125	139,613	15.639	17.938	0.9823	0.9844	0.7979
140	142,811	15.672	17.996	0.9852	0.9875	0.8323
155	144,531	16.168	18.037	0.9871	0.9894	0.8541
170	146,696	15.676	18.084	0.9890	0.9915	0.8792
185	147,696	15.859	18.126	0.9903	0.9929	0.8972
200	147,913	15.652	18.158	0.9916	0.9940	0.9114
215	148,046	15.539	18.194	0.9924	0.9948	0.9212
Proposed	144,312	15.244	17.772	0.9881	0.9916	0.8870
Silent (kb/s)	File size (bytes)	MSU Blocking	MSU blurring	SSIM	3SSIM	stSSIM
130	144,012	11.306	15.865	0.9666	0.9766	0.7385
145	147,470	11.563	16.009	0.9726	0.9817	0.7820
160	149,305	11.236	16.046	0.9763	0.9845	0.8113
175	151,144	11.056	16.105	0.9796	0.9872	0.8374
190	152,906	11.005	16.170	0.9820	0.9901	0.8565
205	153,158	10.792	16.254	0.9842	0.9905	0.8738
220	153,527	10.450	16.267	0.9816	0.9919	0.8887
235	154,385	10.591	16.321	0.9873	0.9926	0.9008
Proposed	152,487	9.873	16.250	0.9812	0.9890	0.8728

Table 2: Encoded file sizes and VQMs for the MPEG-4 dataset. Shaded entries are the closest values to the proposed method in each column. It can be interpreted that the file size and quality of the proposed method are comparable to the video encoded with the bitrate specified in the same row.

[4] R. Feghali, F. Speranza, D. Wang, and A. Vincent. Video quality metric for bit rate control via joint adjustment of quantization and frame rate. *IEEE Transactions on Broadcasting*, 53(1):441–446, 2007.

[5] C. M. Graphics and M. L. V. Group. Msu video quality measurement tool documentation. http://www.compression.ru/video/, 2011.

[6] A. Javdtalab, M. Omidyeganeh, S. Shirmohammandi, and M. Hosseini. A rate control algorithm for x264 high definition video conferencing. In *IEEE International Conference on Multimedia and Expo*, 2011.

[7] R. K. Lama, M. R. Choi, J. Y. Pyun, and G. R. Kwon. Post processing algorithm for reduction of blocking artifact in hdtv. In *IEEE International Conference on Consumer Electronics (ICCE)*, 2014.

[8] C. Li. Content-weighted video quality assessment using a three-component image model. *Journal of Electronic Imaging*, 19(1), 2004.

[9] L. Li, H. Zhu, G. Yang, and J. Qian. Referenceless measure of blocking artifacts by tchebichef kernal analysis. *IEEE Signal Processing Letter*, 21(1):122–125, 2014.

[10] Z. Li and E. J. Delp. Block artifact reduction using a transform-domain markov random field model. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(12):1583–1593, 2005.

[11] A. K. Moorthy and A. C. Bovik. Efficient motion weighted spatio-temporal video ssim index. In *SPIE Proceedings Human Vision and Electronic Imaging*, 2010.

[12] A. Norkin, G. BjÃÿntegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, M. Zhou, and G. V. Auwera. Hevc deblocking filter. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1746–1754, 2013.

[13] C. E. Rhee, J. S. Kim, and H. J. Lee. Bitrate control using a heuristic spatial resolution adjustment for a real-time h.264/avc encoder. *EURASIP Journal on Advances in Signal Processing*, 87, 2012.

[14] G. J. Sullivan, J. R. Ohm, J. W. Han, and T. Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012.

[15] A. M. Tourapis, K. Suhring, and G. Sullivan. H.264/14496-10 avc reference software manual, 2010.

[16] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[17] G. L. Wu, Y. J. Fu, S. C. Huang, and S. Y. Chien. Perceptual quality-regulable video coding system with region-based rate control scheme. In *In Proc. of International Conference of Image Processing*, 2009.

[18] S. B. Yoo, K. Choi, and J. B. Ra. Post-processing for blocking artifact reduction based on inter-block correlation. *IEEE Transactions on Multimedia*, 16(6):1536–1548, 2014.