A Low Power Lossy Frame Memory Recompression Algorithm

Xin Zhou#, Xiaocong Lian*, Wei Zhou*, Zhenyu Liu§, Xiu Zhang*

#School of Automation, Northwestern Polytechnical University, Xi’an, China
*School of Electronics and Information, Northwestern Polytechnical University, Xi’an, China
§Tsinghua National Laboratory for Information Science and Technology, Research Institute of Information Technology, Tsinghua University, Beijing, China
E-mail: zhouwei@nwpu.edu.cn

Abstract—With the development of Ultra-High-Definition video, the power consumed by accessing reference frames in the external DRAM has become the bottleneck for the portable video encoding system design. To reduce the dynamic power of DRAM, a lossy frame memory recompression algorithm is proposed. The compression algorithm is composed of a content-aware adaptive quantization, a multi-mode directional prediction, a dynamic kth-order unary/Exp-Golomb coding and a partition group table based storage space reduction scheme. Experimental results show that, an average data reduction ratio of 71.1% is obtained, while 41% memory space can be saved. 59.6% dynamic power of the DRAM is reduced by our strategies in total. The algorithm causes a controllable video quality degradation, and the BD-PSNR is only -0.04db, or equivalently BD-BR=1.42%.

I. INTRODUCTION

Since entering the 21st century, the video related field is developing rapidly. Video resolution has dramatically increased and Quad Full High Definition (QFHD, 4K) and Super Hi-Vision (SHV, 8K) applications are appearing constantly. The developments of video and network enhance the requirement of a new video coding standard, which has higher coding efficiency and supports the development of the higher-definition video. Therefore, High Efficiency Video Coding (HEVC)[1] was developed by the Joint Collaborative Team on Video Coding(JCT-VC). It can save 40%-50% bitrates compared to H.264/AVC, especially for Ultra-High-Definition video.

The off-chip Dynamic Random Access Memory (DRAM) is applied as frame memory for its high integration density. Bandwidth and power requirements are two deterrences of reference frame storage. By applying the data reusing scheme, the bandwidth limitation can be reduced[2]. Therefore, the power requirement of DRAM has become a more serious bottleneck of portable video codec applications. The dynamic power of DRAM comprises of three primary components, i.e., the internal read/write power, the IO terminal power and the activate power[3]. The first two items can be reduced by saving the IO bandwidth requirements, i.e., improving the data compression ratio, while the last one item can be saved by reducing the memory storage.

Extensive frame memory recompression (FMR) algorithms [4]–[7] have been proposed to reduce the memory bandwidth, which consists of two categories: Lossless and Lossy FMR algorithm. Lossy FMR algorithms utilize the insensitivity of human eyes to small quality loss to further improve the compression efficiency. However, most of the previous lossy algorithms are lack of an effective mechanism to control the encoding quality loss when they adopt the lossy compression to the reference frame in order to get a higher compression ratio.

To overcome the above obstacles, we explore the image context oriented dynamic quantization scheme, which can improve the compression performance on the premise that the image quality is maintained. The simple flow chart of our compression progress is shown in Fig. 1. We first adopt Reference Quantization Length (RQL) Decision method to get the quantization step. Next, we develop the multi-mode directional prediction to obtain the residual r. The quantized residues(r’) are coded with the dynamic kth order unary/Exp-Golomb (UEG) coding algorithm, through a partition group table based storage scheme and stored to the DRAM.

The rest of this paper is organized as follows. In Section II, we introduce the content-aware adaptive quantization method. The proposed multi-mode directional prediction method is explained in Section III. The dynamic kth order unary/Exp-Golomb coding and the partition group table based storage scheme are described in Section IV. Section V illustrates the experimental results. Finally, conclusions are drawn in Section VI.

II. CONTENT-AWARE ADAPTIVE QUANTIZATION

Thoroughly considering the data dependency induced by deblocking filtering and the DRAM burst access properties, the basic luma partition size in our paper is defined as 16×16(16-pixel column by 16-pixel row), and the corresponding chroma partition is 8×8 with 4:2:0 sampling format.

In this section, we will describe the content-aware adaptive quantization. On the basic of the estimated the power of motion estimation residuals, we adjust the quantization step of our lossy off-chip memory compression.
Let $S(u,v)$ represent the discrete 2-D DCT transform coefficients of prediction residuals. During the HEVC coding, the distortion $D$ and the corresponding rate $R$ have the relations expressed as

\[
\begin{align*}
D &= \min(\phi, S_{\alpha}(u,v)) \\
R &= \max(0, \log_2 \frac{S_{\alpha}(u,v)}{\phi})
\end{align*}
\]

where, $S_{\alpha}(u,v)$ is the power spectral density of $S(u,v)$. $\phi$ is the quantization noise, which has the formulation of $\phi = Q^2 / 12$.

In our lossy reference picture compression, the quantization step is $2(l \in \{0,1,2,3\})$. Namely, the last $l$ bits of prediction residuals ($r$), which are generated by the Multi-Mode directional prediction (described in Section II), are discarded. In consequent, the additional noises ($\Delta$) will be introduced to the transformed prediction residuals. Let $\Delta_{ee}$ denote the power spectral density of the additive noise $\Delta$, the corresponding $D$ and $R$ are formulated as

\[
\begin{align*}
\hat{D} &= \min(\phi, S_{\alpha}(u,v) + \Delta_{ee}) \\
\hat{R} &= \max(0, \log_2 \frac{S_{\alpha}(u,v) + \Delta_{ee}}{\phi})
\end{align*}
\]

in which, $\Delta_{ee}$ coming from the reference frame lossy compressing is linear with $2^{2l}$, and is written as $\Delta_{ee} = 2^{2l} / \gamma$. It is assumed that $S_{\alpha}(u,v) + \Delta_{ee} > \phi$ and $S_{\alpha}(u,v) >> \Delta_{ee}$. If the increased rate ($dR = \hat{R} - R$) is desired to be $\beta R$, we can deduce the value of $l$ as

\[
l = \frac{1}{2} \log_2 (\alpha S_{\alpha}(u,v) \log_2 \frac{12S_{\alpha}(u,v)}{Q^2})
\]

in which $\alpha = \beta \cdot \gamma / 2$. It can be observed that $l$ increases with the prediction residual power $S_{\alpha}(u,v)$. Therefore, the essential of our paper is to determine $l$ from the estimation of $S_{\alpha}(u,v)$ and the experimental parameter $\alpha$. $\alpha$ is defined from the experiments, which will be described in Section V. The value of $S_{\alpha}(u,v)$ depends on the future Motion Estimation using the current block as the reference. Based on the related analysis[8], $S_{\alpha}(u,v)$ can be estimated from the textures and motions of the current 16×16 block. We divided the 16×16 block into sixty-four 2×2 sub-blocks, of which the edge vectors are calculated as follows.

\[
\begin{align*}
dx_{i,j} &= p_{2i+1,2j} + p_{2i+1,2j+1} - p_{2i+2,2j} - p_{2i,2j+1} \\
dy_{i,j} &= p_{2i,2j+1} + p_{2i+1,2j+1} - p_{2i+2,2j+1} - p_{2i+1,2j} \\
\end{align*}
\]

where $p_{i,j}$ is the pixel value, $dx_{i,j}$ and $dy_{i,j}$ represent the edge strength in the vertical and horizontal directions of 2×2 block, respectively.

The motion vectors ($\overline{MV}_{ij} = \{MV_{x_{ij}}, MV_{y_{ij}}\}$) are derived by the results of the HEVC inter encoder. Meanwhile, in order to strengthen the effect of motion vector, we introduce two variables $\Theta_x$ and $\Theta_y$, which are derived as follows.

\[
\begin{align*}
\Theta_x &= (MV_x > 8) \rightarrow 2:1 \\
\Theta_y &= (MV_y > 8) \rightarrow 2:1
\end{align*}
\]

Therefore, the approximate value of $S_{ee}(u,v)$, i.e., $\tilde{S}_{ee}(u,v)$ is estimated as

\[
\tilde{S}_{ee}(u,v) = \frac{1}{1024} \sum_{i,j=0}^{8} ((dx_{i,j} MV_{x})^2 \Theta_x + (dy_{i,j} MV_{y})^2 \Theta_y)
\]

Based on the above equations, we can calculate $l$. The lowest $l$ bits of the prediction residuals are discarded. To maintain the image quality, if $\tilde{S}_{ee}(u,v) < \frac{1}{6} Q^2$, we set $l=0$.

III. MULTI-MODE DIRECTIONAL PREDICTION

This paper proposes a multi-mode directional prediction to improve the prediction accuracy. We estimate the edge direction of current pixel from its left and top 2×2 blocks, as shown by Fig. 2. In order to adapt to the different applications, we design three prediction modes (mode 0, mode 1, and mode 2, as shown in Fig. 2), which can predict one, two, and four pixels at a time, respectively. For the proposed three prediction mode, the throughput is enhanced at the cost of the prediction accuracy loss.

Fig. 1. Pixel locations in one 16×16 partition and corresponding prediction modes
For the current pixel \( p_{ij} \) \((p_{i,j+1}, p_{i+1,j}, p_{i+1,j+1})\), we define the corresponding left and top neighboring edge vectors as, \( \hat{D}_{ij} = \{dx_{ij},dy_{ij}\} \) and \( \hat{D}_{ij} = \{dx'_{ij},dy'_{ij}\} \), respectively, and the block with the larger strength value is denoted as the final reference candidate, i.e., \( \hat{D}_{ij} \). While for the blue and yellow region, they only have one prediction block. In \( \hat{D}_{ij} \), \( dx_{ij} \) and \( dy_{ij} \) represent the edge strength in vertical and horizontal directions, respectively. The edge direction of current pixel can be estimated by the ratio between \( , \) and \( , \) respectively.

The edge direction of current pixel \( \eta \), is calculated based on the value of \( \eta \), which is shown in (7).

\[
\begin{align*}
\theta &= \begin{cases} 
45^\circ & \text{if } 0.5 < \eta(\hat{D}_{ij}) \leq 2 \\
67.5^\circ & \text{if } 2 < \eta(\hat{D}_{ij}) \leq 4 \\
90^\circ & \text{if } \eta(\hat{D}_{ij}) > 4 \\
112.5^\circ & \text{if } -4 \leq \eta(\hat{D}_{ij}) \leq -2 \\
135^\circ & \text{if } -2 < \eta(\hat{D}_{ij}) \leq -1 \\
157.5^\circ & \text{if } -1 < \eta(\hat{D}_{ij}) \leq -0.25 \\
180^\circ & \text{if } -0.25 < \eta(\hat{D}_{ij}) \leq 0.5 
\end{cases}
\end{align*}
\]

(7)

The \( 45^\circ, 90^\circ, 135^\circ, 180^\circ \) directions use the copy of pixels to get the residuals, whereas the others get the prediction by calculating the average of two neighboring pixels. For the pixels in the last column, as the reference pixel on the top-right does not exit, the directions \( 45^\circ \) and \( 67.5^\circ \) are skipped.

IV. ENTROPY CODING AND STORAGE SCHEME

The adaptive order unary/Exp-Golomb rice coding is used as our entropy coding algorithm. After inputting the value \( x \), we get the quotient \( q = x / 2^\alpha \) and the remainder \( r = x \% 2^\alpha \). For the quotient part, we use the improved unary/Exp-Golomb coding with threshold 3. That is, when \( q < 3 \), we use the unary coding; otherwise, the improved Exp-Golomb coding is applied. The reminder \( r \) is stored followed the coded quotient. If \( x \neq 0 \), the sign bit need to be stored, which is transmitted after the remainder part. In order to obtain the optimal order \( k_0 \), we also use the directional prediction method to realize the efficient pixel-grain \( k \) order update [9].

Many all-zero-residual areas are found in the chroma partition, and it has a great potential to further improve the compression ratio. Two compression skip flags, i.e., partition compression skip flag (PCSF) and block compression skip flag (BCSF) are proposed in our paper to mark the all-zero region.

Because of the unfixed DRR of our algorithm, the compressed partitions can not be addressed linearly. Therefore, the partition group table based storage method is proposed in this paper to resolve the address mapping issues. In detail, every two horizontal adjacent \( 16 \times 16 \) partitions compose one partition group(PG). One dedicated PG table is assigned to each PG, which contains the auxiliary information of the two partitions used for decompression. When the DRR of luma and chroma components of two partitions are all no less than 50%, the content of one PG can be stored into the space of one partition; Otherwise, two-partition space are still needed. The memory storage mapping and the auxiliary information are shown in Fig.3. By applying the proposed storage scheme, our algorithm can not only reduce the bandwidth requirements, but also save the memory space of DRAM, and then save the power requirements.

V. EXPERIMENTS RESULTS

The proposed method is conducted on HEVC reference test model HM15.0. In this experiment, ten typical video sequences in class A-E with QP = \( (22, 27, 32, 37) \) are tested to analyse the compression performance.

The variable \( \alpha \) is an important factor which affects the quantization result. Four sets of experiments with different \( \alpha \) are tested to analyze the compression performance of Content-Aware Adaptive Quantization and results are shown in Table I. We can see from the experiments that it can get a best compression performance when \( \alpha = 0.0002 \). The image quality loss in terms of BD-PSNR is only -0.044db.

<table>
<thead>
<tr>
<th>Table I. COMPRESSION PERFORMANCE OF CONTENT-AWARE ADAPTIVE QUANTIZATION(QP=22,27,32,37, MODE = 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha )</td>
</tr>
<tr>
<td>BD-PSNR(db)</td>
</tr>
<tr>
<td>BD-BR(%)</td>
</tr>
<tr>
<td>DRR(%)</td>
</tr>
</tbody>
</table>

In order to apply to the different situations, we propose three prediction modes which can predict one, two and four pixels in each cycle. The compression performance of them is shown in Table II. Mode 0 is a method which can get the highest compression ratio, but the throughput is low as it can only handle one pixel at a time. Mode 1 is an effective method which can double the throughput with only 1.25% DRR decrease. Mode 2 can get a quadruple throughput with a DRR decrease of 5.67%. We can make a balance between the compression performance and the throughput in accordance to
the specific application scenario.

<table>
<thead>
<tr>
<th>class</th>
<th>video sequence</th>
<th>Mode 0(%)</th>
<th>Mode 1(%)</th>
<th>Mode 2(%)</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>PeopleOnStreet</td>
<td>68.86</td>
<td>67.51</td>
<td>63.27</td>
<td>-1.03</td>
</tr>
<tr>
<td>B</td>
<td>BasketballDrive</td>
<td>77.04</td>
<td>76.73</td>
<td>71.50</td>
<td>-0.10</td>
</tr>
<tr>
<td>C</td>
<td>BasketballPass</td>
<td>69.98</td>
<td>68.63</td>
<td>63.34</td>
<td>-0.01</td>
</tr>
<tr>
<td>D</td>
<td>Johnny</td>
<td>78.47</td>
<td>77.65</td>
<td>72.12</td>
<td>-0.04</td>
</tr>
<tr>
<td>E</td>
<td>KristenAndSara</td>
<td>71.07</td>
<td>69.82</td>
<td>65.40</td>
<td></td>
</tr>
</tbody>
</table>

Our reference frame recompression algorithm also contributes to DRAM power reduction, and the CACTI simulator\[8\] is introduced to measure the dynamic power of DRAM with the 1.5V core and IO voltages. The internal read/write power, the IO terminal power and the activate power consume 40.1%, 21.7% and 38.2% of the total dynamic power \[10\], respectively. On average, 41% of the memory space is reduced by the proposed structure. Accordingly, 15.7% DRAM dynamic power is saved in our experiments.

Table III shows the comparisons of proposed algorithm with previous algorithms. Lee’s \[5\] work can get a fixed DRR of 50%, and the memory storage is also saved by half. But it caused a large image quality loss in order to get the fixed compression ratio. A DRR of 65.3% was obtained in Cheng’s \[6\] work, and the image quality loss was -0.10db. Because of the high computational complexity, its target resolution is only 352 × 288. Fan’s \[7\] algorithm achieved 67% bandwidth reduction by truncating the lower 3 bits of the pixels for the IME part, but for the FME and MC parts, the truncated 3 bits were added and the bandwidth reduction was only 29.4%.

<table>
<thead>
<tr>
<th>ALGORITHMS</th>
<th>DRR(%)</th>
<th>Space save(%)</th>
<th>Power save(%)</th>
<th>Complexity</th>
<th>Target Resolution</th>
<th>BD-PSNR(db)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lee’s[5]</td>
<td>50.0(fixed)</td>
<td>50</td>
<td>50.0</td>
<td>Low</td>
<td>1080p</td>
<td>-1.03</td>
</tr>
<tr>
<td>Cheng’s[6]</td>
<td>65.3</td>
<td>0</td>
<td>40.4</td>
<td>Medium</td>
<td>16×16</td>
<td>-0.10</td>
</tr>
<tr>
<td>Fan’s[7]</td>
<td>54.6</td>
<td>0</td>
<td>41.4/18.2</td>
<td>High</td>
<td>8×8</td>
<td>-0.01</td>
</tr>
<tr>
<td>Proposed</td>
<td>71.1</td>
<td>41</td>
<td>59.6</td>
<td>Medium</td>
<td>16×16</td>
<td>-0.04</td>
</tr>
</tbody>
</table>

VI. CONCLUSION

In order to reduce the memory bandwidth and power requirements, a lossy FMR algorithm is proposed in this paper. The algorithm consists of four parts, a content-aware adaptive quantization, a multi-mode directional prediction, a dynamic k-th-order unary/Exp-Golomb coding and a partition group table-based storage space reduction scheme. The experimental results demonstrate that, 71.1% memory bandwidth and 41% memory space are saved by our algorithm, and 59.6% DRAM dynamic power is reduced in total. The video quality loss caused by our algorithm is -0.04db in terms of BD-PSNR.

ACKNOWLEDGEMENT

This work was supported in part by the the National Natural Science Foundation of China (61602383), Fundamental Research Funds for the Central Universities (3102016Z024), Fundamental Natural Science Research Funds of Shaanxi Province (2016JM6077).

REFERENCES