VLSI ORIENTED FAST MULTIPLE REFERENCE FRAME MOTION ESTIMATION ALGORITHM FOR H.264/AVC

Zhenyu Liu, Lingfeng Li, Yang Song, Takeshi Ikenaga, Satoshi Goto

The Graduate School of IPS, Waseda University
N355, 2-7, Hibikino, Wakamatsu, Kitakyushu, 808-0135, Japan

ABSTRACT

In H.264/AVC standard, motion estimation can be processed on multiple reference frames (MRF) to improve the video coding performance. For the VLSI real-time encoder, the heavy computation of fractional motion estimation (FME) makes the integer motion estimation (IME) and FME must be scheduled in two macro block (MB) pipeline stages, which makes many fast MRF algorithms inefficient for the computation reduction. In this paper, two algorithms are provided to reduce the computation of FME and IME. First, through analyzing the block’s Hadamard transform coefficients, all-zero case after quantization can be accurately detected. The FME processing in the remaining frames for the block, detected as all-zero one, can be eliminated. Second, because the fast motion object blurs its edges in image, the effect of MRF to aliasing is weakened. The first reference frame is enough for fast motion MBs and MRF is just processed on those slow motion MBs with a small search range. The computation of IME is also highly reduced with this algorithm. Experimental results show that 61.4%-76.7% computation can be saved with the similar coding quality as the reference software. Moreover, the provided fast algorithms can be combined with fast block matching algorithms to further improve the performance.

1. INTRODUCTION

The superior performance of the latest international video coding standard, H.264/AVC, mainly comes from the new techniques, which include 1/4-pixel accurate variable block size motion estimation (VBS-MSE) with multiple reference frames (MRF), intra prediction (IP), context-based adaptive variable length entropy coding (EC) and in-loop deblocking (DB), etc. The huge computation complexity of the prediction algorithm makes the traditional 2-stage MB pipeline architecture inefficient for the H.264 hardwired encoder design. For example, in H.264, FME is 100 times more complex than that of previous standards [1]. It has become the system bottleneck because of the 1/4-pixel accuracy, VBS, MRF and precise distortion evaluation. Consequently, IME and FME must be arranged in two stages to get the high hardware utilization and throughput. One optimized 4-stage pipeline architecture is also introduced here. In the first MB stage, IME engine is processed on all reference frames. The integer motion vectors (MV) of 41 blocks in MB on all reference frames are achieved and dispatched to the second FME stage. Through 1/4-pixel accurate fine ME and precise RD-cost evaluation, FME engine finds the best candidates and the corresponding reference frames and decides the best inter prediction mode. The post inter/intra mode decision IP and chroma MC are implemented at the third stage. EC and DB are processed in parallel in the fourth stage.

According to the analysis in [2], 89.2% computation power is consumed by ME part. MRF is the main issue that leads to the huge computation complexity. In order to reduce the computation, many studies and excellent works about fast MRF ME have been proposed. One excellent work is provided in [2], which provides four criterions to early terminate the motion search on MRFs. These algorithms efficiently reduce 30%-80% redundant computation in the software. However, MB pipeline in hardware architecture degrade the performance of these methods. All these criterions must be applied in the second FME stage. That means the computation load of IME, which is the most computation intensive part, can not be saved. Another promising scheme is reducing the search areas on MRFs depending on the MVs’ strong correlations in consequent pictures [3][4]. The first drawback of this algorithm is the hardware overhead consumed by the MV composition. For example, in reference [3], 4×4 block based MVSs on each frame must be kept. If the frame size is 720×480 with 128×128 search range and 5 reference frames, totally 1.65Mb memories are required. For the accuracy of MV composition, the multiplication, which increases the hardware cost, is also applied. Moreover, this algorithm just simplifies the computation of IME and does not benefit FME engine.

Based on the 4-stage pipeline hardware architecture, two algorithms are proposed in this paper. The first one is Hadamard transform coefficients based all-zero block detection, which can more precisely detect all-zero blocks before real DCT transform and quantization than previous algorithms. The second is the MV based reference frame elimination and search area adjustment algorithm. In details, for the fast motion MB, just one reference frame is adopted; for the slow motion MB, multiple reference frames are processed, but the search area for other frames can be greatly reduced. This approach contributes to the computation reduction of both IME and FME engines.

The rest of this paper is organized as follows. In Section 2, Hadamard transform coefficients based all-zero block detection algorithm is proposed. The motion vector based reference frame elimination and search area adjustment method is present in Section 3.

This work was supported by fund from MEXT via Kitakyushu innovative cluster projects and CREST, JST.
Section 4 shows some experimental results to demonstrate our algorithms. Conclusions are drawn in Section 5.

2. HADAMARD TRANSFORM COEFFICIENTS BASED ALL-ZERO BLOCK DETECTION ALGORITHM

In H.264 standard, after the prediction procedure, the residual blocks are DCT transformed, quantized and entropy coded. The separable 4×4 2-D DCT in H.264 is written as (1).

\[ Y = CXC^T \] (1)

Where the superscript \( T \) denotes transposition, \( X \) is the 4×4 residual matrix and \( C \) is the transform matrix, as shown in (2).

\[
C = \begin{bmatrix}
C_0 & 1 & 1 & 1 \\
C_1 & 1 & -1 & -2 \\
C_2 & 1 & -1 & 1 \\
C_3 & 1 & 2 & -1 \\
\end{bmatrix}
\] (2)

During the quantization step, the thresholds in 4×4 matrix \( \text{TH_{DCT}} \) are shown in (3).

\[
\text{th}_{\text{DCT}}[QP][i][j] = \frac{2^q\text{bits} - q\text{p}_\text{const}}{\text{quant}_{\text{coe}} / q\text{p}_\text{rem}[i][j]} \quad i, j \in [0, 3] \] (3)

where \( QP \) is the quantization parameter, \( q\text{p}_\text{rem} = QP/64, q\text{p}_\text{bits} = QP/6+15, q\text{p}_\text{const} = (1 << q\text{p}_\text{bits})/6 \) and \( \text{quant}_{\text{coe}} \) is the scaling matrix shown in [5].

For one block, during its MRF ME, if we found its residues at current reference frame are small enough to make it all-zeros post transformation (DCT) and quantization (Q), its search processing can be terminated. Depending on its SAD or SATD value, reference [2] has provided one smart method for the early estimation of the all-zero block. However, SAD and SATD cannot emulate the frequency characteristics, so they just provide the coarse all-zero estimations.

In the JM, SATD is applied in the fractional pixel motion estimation and the mode decision, because SATD accounts for the amount of prediction error, as well as the cost for transformed representation. Hence, the SATD operates as one more accurate RD cost criterion for the prediction error than SAD. This algorithm has already been implemented in the hardware design [1]. For SATD calculation, the 4×4 Hadamard transform is applied to each 4×4 residual block as shown in (4). Depending on these Hadamard coefficients, we can get more accurate all-zero estimations.

\[
Z = HXH^T
\] (4)

Where \( H \) is the Hadamard transform matrix, as shown in (5).

\[
H = \begin{bmatrix}
H_0 & 1 & 1 & 1 \\
H_1 & 1 & -1 & -1 \\
H_2 & 1 & -1 & 1 \\
H_3 & 1 & 1 & -1 \\
\end{bmatrix}
\] (5)

Comparing \( H \) and \( C \), we find that the basic functions of \( H \), \( H_0 \) and \( H_2 \), are the same as \( C_0 \) and \( C_2 \) of \( C \) and \( H_1 \) and \( H_3 \) have the similar patterns as \( C_1 \) and \( C_3 \). In fact, Hadamard transform is a simplified formation of DCT transform. Its resulting transformed signal emulates the frequency characteristics of the true DCT transformed block in the subsequent DCT/Q stage at very low computational cost.

Since we have already derive the Hadamard coefficients during FME, a threshold matrix \( \text{TH}_{H} \) can be set for the early detection of all-zero block. The entries in \( \text{TH}_{H} \) have the same value as illustrated in (6). If the Hadamard coefficient is less than the threshold, it is assumed that the corresponding DCT coefficient will become zero after the quantization stage.

\[
\text{th}_{H}[QP][i][j] = \text{th}_{\text{DCT}}[QP][0][0]
\] (6)

This setting depends on two reasons. First, for \( (i, j) = \{ (0, 0), (0, 2), (2, 0), (2, 2) \} \), \( z(i, j) = y(i, j) \) and \( \text{th}_{H}[QP][i][j] = \text{th}_{\text{DCT}}[QP][i][j] \). So we can get the real values of these entries post DCT/Q stage. Second, in other entries, the ratio of the DCT coefficient standard deviation to the Hadamard coefficient standard deviation is similar to that of their thresholds.

Proof: According to [2], the standard deviation matrix of DCT and Hadamard coefficients can be expressed as (7) and (8) respectively, where \( \sigma_f \) denotes the standard deviation of residues. The ratio matrix of \( \sigma_{DCT} / \sigma_{H} \) is illustrated in (9). If \( q\text{p}_\text{const} \) item is eliminated in (3) to simplify the analysis, the ratio of \( \text{TH}_{DCT} \) to \( \text{TH}_{H} \), which is denoted as \( R_{TH} \), has 6 cases according to \( q\text{p}_\text{rem} \). For example, for \( q\text{p}_\text{rem} = 0 \), \( R_{TH} \) is shown as (10) and the ratio between \( R_{TH} \) and \( R_{H} \) is shown in (11). Other cases can be traced by analogy.

\[
\begin{align*}
\sigma_{DCT} & = \sigma_f \\
\sigma_{H} & = \sigma_f \\
R_{H} & = \sigma_{DCT} \odot \sigma_{H} \\
R_{TH} & = \text{TH}_{DCT} \odot \text{TH}_{H} \\
R_{TH} \odot R_{H} & = \begin{bmatrix}
1.00 & 1.63 & 1.00 & 1.63 \\
1.63 & 2.50 & 1.63 & 2.50 \\
1.00 & 1.63 & 1.00 & 1.63 \\
1.63 & 2.50 & 1.63 & 2.50 \\
\end{bmatrix}
\end{align*}
\] (7)

where \( \odot \) is the scalar division, which means each entry of the first matrix is divided by the element in the same position in the second matrix.

According to the above analysis, two early termination criteria are provided to alleviate the computation load in FME stage:

1. if((mode == P16×16) && (P16×16_block == all_zero_block) && (MV == SKIP_MV)), early terminate.
2. if(current_block == all_zero_block), early terminate.

Through analyzing its Hadamard coefficients, we can make early all_zero_block detection for each block. If the current block mode is P16×16 and it is decided as all_zero_block at its SKIP_MV search position, this MB is set as SKIP_MODE and all other searches are eliminated. For other blocks, once they are estimated as all_zero_block case, their searches on subsequent reference frames are terminated. The effect of this algorithm depends on QP value. For QP is 32, 21%-46% FME calculation can be saved.
Since the Hadamard transformation module has already been
built in the FME engine [1], according to the 2-D Hadamard archi-
tecture, 4 comparators are required to be added to the output of each
2-D 4×4 Hadamard module for the all-zero block detection. This
additional hardware overhead is trivial compared with the FME en-
gine.

3. MOTION VECTOR BASED REFERENCE FRAME
ELIMINATION AND SEARCH AREA ADJUSTMENT

By mathematical analysis, aliasing is the main component that dis-
termines the prediction efficiency. Sub-pel interpolation and MRF
techniques adopted by H.264 mainly aim to compensate the aliasing.
In this section, first, the prediction error signal caused by aliasing is
introduced. Second, the effect of image motion to aliasing is analyt-
ically described. At last, our MV based reference frame elimination
and search area adjustment algorithm is proposed.

3.1. Impact of Aliasing to Prediction Error Signal

In order to simplify the mathematical description, the analysis is re-
stricted to one spatial dimension signal. lₜ(x) and lₜ₋₁(x) denote the
spatial-continuous signals at time instance t and t − 1. lₜ(x) is a dis-
placed version of lₜ₋₁(x) and the distance is dₛ, which can be ex-
pressed as lₜ(x) = lₜ₋₁(x − dₛ). Their frequency domain signals
are denoted as Lₜ(ω) and Lₜ₋₁(ω). These continuous image sig-
nals are sampled by the sensor array before digital processing and
they are denoted as xₜ(xₙ₁) and xₜ₋₁(xₙ₁). Aliasing does not ex-
ist if Nyquist-Shannon sampling preconditions, i.e., Lₜ₋₁(ω) = 0 for
|ω| ≥ ωₛ/2, where ωₛ is the sampling frequency, is satisfied. How-
ever, because no time-limited signals can be band limited and the
low-pass filter of the sampling system is not ideal, the preconditions
of Nyquist-Shannon sampling theorem cannot be fulfilled.

According to [6], with the normalized sampling frequency, i.e.,
ωₛ = 2π, the magnitude of prediction error signal caused by aliasing
can be described as (12)

|Eₜ(ω)| = 2 ∙ |Aₜ₋₁(ω)| ∙ |sin(dₛ ∙ ωₛ)| (12)

where Aₜ₋₁(ω) = Lₜ₋₁(ω + 2π) + Lₜ₋₁(ω − 2π).

According to (12), two important conclusions can be drawn:

1. Because of the item |Aₜ₋₁(ω)|, aliasing is caused by the high
frequency signals in Lₜ₋₁(ω), where |ω| ≥ π.

2. According to the item |sin(dₛ ∙ ωₛ)|, the impact of aliasing
vanishes at full pixel displacements and is maximum at half
pixel displacements.

Conclusion 1 states that the image rich of high frequency sig-
nals is prone to be affected by the aliasing problem. Conclusion 2
explains the necessity of MRF during prediction processing: If the
displacement dₛ,ₜ₋₁ between the current xₜ(xₙ₁) and the previous
xₜ₋₁(xₙ₁) image is sub-pel and the more previous image xₜ₋₄(xₙ₁)
has the full-pel displacement dₛ₋₄,ₜ₋₄, xₜ₋₄(xₙ₁) is preferred to be
chosen as reference because the aliasing problem does not exist any
more.

Now, we can explain why ‘Mobile’ sequence is so sensitive to
MRF. Many textures are contained in this video sequence. Sharp
edges in the spatial domain generate rich high frequency signals after
the Fourier transformation. For example, flickering at the edges of
calender is caused by input aliasing. Though 2-D Wiener filter inter-
polation algorithm in H.264/AVC can alleviate the error of aliasing,
its effect can not compare with the reference image with the full-pel
displacement. In fact, through our experiments, aliasing is the main
reason for MRF.

3.2. Effect of Image Motion to Multiple Reference Frame Algo-

In the first reference frame, VBSME is processed in the full
search range, S_FW × S_FH.

2. If the |MVₓ| + |MVᵧ| of P16 × 16 is beyond the threshold
TH_MV, that means that current MB is a fast motion one,
### Table 1. Simulation conditions

<table>
<thead>
<tr>
<th>QP</th>
<th>Search</th>
<th>Range</th>
<th>etc</th>
</tr>
</thead>
<tbody>
<tr>
<td>16, 20, 24, 28, 32</td>
<td>Jm81a</td>
<td>±16</td>
<td>no B slice, CA VLC, 5-ref, RDO, Off, Hadamard Transform</td>
</tr>
</tbody>
</table>

### Table 2. Coding speed-up ratio

<table>
<thead>
<tr>
<th>QP</th>
<th>16</th>
<th>20</th>
<th>24</th>
<th>28</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreman</td>
<td>2.68</td>
<td>2.75</td>
<td>2.96</td>
<td>3.33</td>
<td>3.94</td>
</tr>
<tr>
<td>Carphone</td>
<td>2.77</td>
<td>2.89</td>
<td>3.37</td>
<td>3.58</td>
<td>4.11</td>
</tr>
<tr>
<td>Mobile</td>
<td>2.63</td>
<td>2.59</td>
<td>2.68</td>
<td>2.82</td>
<td>3.07</td>
</tr>
<tr>
<td>Coastguard</td>
<td>3.47</td>
<td>3.42</td>
<td>3.43</td>
<td>3.70</td>
<td>3.93</td>
</tr>
<tr>
<td>Football</td>
<td>3.43</td>
<td>3.42</td>
<td>3.54</td>
<td>3.58</td>
<td>3.80</td>
</tr>
<tr>
<td>Tempe</td>
<td>3.11</td>
<td>3.06</td>
<td>3.01</td>
<td>3.11</td>
<td>3.23</td>
</tr>
</tbody>
</table>

so the searches on subsequent reference frames can be eliminated. Otherwise, as current MB is the slow motion one and aliasing exists, MRF is still required. However, VB-SME on other frames can be processed in the reduced ranges $S_{SW} \times S_{SH}$ to save the computation of IME part.

As small blocks contain less texture and prone to be trapped in local optimum positions, the motion feature judgment just depends on the MV of P16 $\times$ 16 block. Moreover, this approach also simplifies its computation complexity. This criterion is placed on the first IME stage in Figure 1. If current MB is decided as fast motion type, the first IME engine and second stage FME engine just have one reference frame. In this way, the computation load of IME and FME are both saved. For those slow motion MBs, this algorithm just contributes to the IME’s computation reduction.

### 4. EXPERIMENTAL RESULTS

Six standard sequences at QCIF format and 30Hz, Forman (255 frames), Carphone (255 frames), Mobile (249 frames), Coastguard (255 frames), Football (249 frames) and Tempe (249 frames), are tested to compare bit-rate, PSNR and coding speed. Among these sequences, Forman, Carphone and Mobile are the most sensitive to the reference frame number. Other simulation conditions are shown in Table 1. The RD-curve comparisons of the six test vectors are shown in Figure 3. As our algorithms provide the almost the same coding efficiency, it is hard to distinguish our algorithms’ curves from the reference ones.

The experimental coding-speedup ratio results are listed in Table 2. The coding-speedup ratio is defined as the ratio of the full-search algorithm’s coding time to the one of ours in the case of 5 reference frames. Table 2 demonstrates that 61.4% to 76.7% computation can be save by our schemes. We can see that the ratio is commonly increased with QP. This mainly comes from the enhancement of all-zero block detection algorithm’s effect with the incensement of QP.

It should be noticed that our algorithms are orthogonal to fast block matching algorithms, namely, if fast block matching algorithms are applied in the first reference frame search, more computation can be saved.

### 5. CONCLUSIONS

Fully considering the limitations of MB-pipeline hardware architectures, we propose two VLSI friendly fast algorithms for MRF ME in H.264/AVC: Hadamard transform coefficients based all-zero block detection algorithm efficiently alleviates the computation load of FME part; The effect of motion to aliasing on the prediction error is theoretically and experimentally investigated. The MV based reference frame elimination and search area adjustment algorithm is proposed to save operations in both IME and FME parts. Experimental results show that 61.4%-76.7% computation can be saved with almost the same coding quality as the reference software. Moreover, these provided schemes can be combined with other fast block ME algorithms to further improve the performance.

### 6. REFERENCES


