Research Article

Improved Motion Estimation Using Early Zero-Block Detection

Y. M. Lee, Y. J. Tsai, and Y. Lin

Department of Communication Engineering, National Central University, Chungli 32054, Taiwan

Correspondence should be addressed to Y. M. Lee, yuming0727@gmail.com

We incorporate the early zero-block detection technique into the UMHexagonS algorithm, which has already been adopted in H.264/AVC JM reference software, to speed up the motion estimation process. A nearly sufficient condition is derived for early zero-block detection. Although the conventional early zero-block detection method can achieve significant improvement in computation reduction, the PSNR loss, to whatever extent, is not negligible especially for high quantization parameter (QP) or low bit-rate coding. This paper modifies the UMHexagonS algorithm with the early zero-block detection technique to improve its coding performance. The experimental results reveal that the improved UMHexagonS algorithm greatly reduces computation while maintaining very high coding efficiency.

1. INTRODUCTION

The newest international video coding standard H.264/AVC has recently been approved by the ITU-T (as recommendation H.264) and by ISO/IEC as the international standard MPEG-4 part 10 advanced video coding (AVC) standard [1]. The emerging H.264/AVC achieves significantly better performance in both PSNR and visual quality at the same bit-rate compared with prior video coding standards such as MPEG4 part 2 and H.263. One important technique is the use of the variable block-size motion estimation and rate distortion optimization techniques; the computational complexity of H.264/AVC is dramatically increased due to the variable block-size modes performed.

Many fast and efficient methods for motion estimation (ME) have been proposed in recent years to reduce computational cost and maintain coding performance. In general, there are two ways to reduce computation. One is to speed up the ME algorithms themselves, such as the hybrid unsymmetrical-cross multihexagon-grid search (UMHexagonS) algorithm [2], which has been adopted in JM reference software. The other is to terminate the ME calculation by early detection of the zero-blocks (ZBs) of discrete cosine transform (DCT) coefficients after quantization.

Xie et al. [3] established a zero-block condition based on the following criterion:

$$|X(0,0)| \leq Q_{step},$$

$$\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x^2(i,j) - X^2(0,0) \leq \frac{N^2}{12} \times (Q_{step})^2, \quad (1)$$

where $X(0,0) = (1/N)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} x(i,j)$, and $x(i,j)$ is residual samples between the current macroblock and the reference macroblock. For H.264, the relation between Q_{step} and quantization parameter is $Q_{step} = 0.625 \cdot 2^{QP/6}$. This criterion has been employed in the JM reference software. In [4, 5], the early zero-block detection approach was applied to the motion search process using a threshold of $20Q_{step}$ for comparison with the sum of difference of 8×8 block size ($SAD_{8\times8}$) and deciding whether 8×8 DCT is a zero-block. The motion search stops when all zero-blocks are detected. This results in significant computational savings, especially for low bit-rate coding. The threshold of $20Q_{step}$ (corresponding to $5Q_{step}$ in 4×4 discrete cosine transform and quantization (DCT/Q)) is not sufficient, and it could improperly detect a great number of zero-blocks, leading to a severe degradation in coding performance.

Some sufficient but not necessary conditions for zero-block detection of DCT coefficients after quantization were derived by examining the sum of absolute differences (SADs) between the current macroblock and the reference macroblock [6, 7]. Although the zero-blocks of DCT coefficients can be detected correctly, numerous zero-blocks still remain undetected. Based on Moon's method [7], a technique using an adaptive threshold was suggested to enhance zero-block detecting capability [8].

In this work, we derive a nearly sufficient condition based on the ensemble average of all 4×4 DCT coefficients. The nearly sufficient condition for zero-block detection is then applied to both motion search and DCT/Q calculation in the UMHexagonS algorithm. The experimental results reveal that a significant improvement in computation reduction can be achieved compared to methods using the other two sufficient conditions, while high coding efficiency is still maintained.

2. A NEARLY SUFFICIENT CONDITION FOR ZERO-BLOCK DETECTION

To guarantee integer transform, the 4×4 DCT in H.264/AVC is approximated to the following form:

$$Y = CXC^T \otimes \text{PF}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \cdot [X] \cdot \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix}$$

$$\otimes \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix},$$

$$\tag{2}$$

where $a = 1/2$, $b = \sqrt{2/5}$, and $c = 1/2$. The basic quantization operation is given by

$$Z_{ij} = \text{round}\left(\frac{Y_{ij}}{Q_{\text{step}}}\right). \tag{3}$$

The value of quantization parameter (QP) varies in the range 0–51. The quantizer step size Q_{step} is used to control bit-rate and video quality. With postscaling factor (PF) considered with the quantizer, the quantized output Z_{ij} can be written as

$$Z_{ij} = \text{round}\left(W_{ij} \cdot \frac{\text{PF}}{Q_{\text{step}}}\right), \tag{4}$$

where W_{ij} is the entry of the core 2D transform $W = CXC^T$. To avoid any division operation, the factor $(\text{PF}/Q_{\text{step}})$ is implemented by a multiplication factor and a right shift:

$$Z_{ij} = \text{round}\left(W_{ij} \cdot \frac{M(\text{QP}\%6; r)}{2^{q\text{bit}}}\right),$$

$$\tag{5}$$

$$\frac{\text{PF}}{Q_{\text{step}}} = \frac{M(\text{QP}\%6; r)}{2^{q\text{bit}}}$$

with

$$M(\text{QP}\%6; r) = \begin{bmatrix} 5243 & 8066 & 13107 \\ 4660 & 7490 & 11916 \\ 4197 & 6554 & 10082 \\ 3647 & 5825 & 9362 \\ 3355 & 5243 & 8192 \\ 2893 & 4559 & 7282 \end{bmatrix}, \quad r = 0, 1, 2,$$

$$\tag{6}$$

$$q\text{bit} = 15 + \left\lfloor \frac{\text{QP}}{6} \right\rfloor,$$

where $r = 2 - (i\%2) - (j\%2)$, % denotes the modular operator, and $M(\text{QP}\%6; r)$ is the multiplication factor. The quantized coefficient can be implemented using integer arithmetic:

$$Z_{ij} = (W_{ij} \cdot M(\text{QP}\%6; r) + f) \gg q\text{bits}, \tag{7}$$

where $\gg$ represents a binary shift right, and f is $2^{q\text{bit}}/6$ for interblocks or $2^{q\text{bit}}/3$ for intrablocks.

Sousa [6] derived a simple sufficient condition under which each quantized coefficient becomes zero for 8×8 DCT. To derive the sufficient condition for 4×4 DCT, the PF factor is absorbed back into the core 2D transform and 4×4 DCT coefficients are rewritten Y:

$$Y = CXC^T \otimes \text{PF} = DXD^T$$

$$= \begin{bmatrix} a & a & a & a \\ b & \tilde{c} & -\tilde{c} & -b \\ a & -a & -a & a \\ \tilde{c} & -b & b & -\tilde{c} \end{bmatrix} \cdot [X] \cdot \begin{bmatrix} a & b & a & \tilde{c} \\ a & \tilde{c} & -a & -b \\ a & -\tilde{c} & -a & b \\ a & -b & a & -\tilde{c} \end{bmatrix}, \tag{8}$$

where $a = 1/2$, $b = \sqrt{2/5}$, and $\tilde{c} = (1/2)\sqrt{2/5}$. Each coefficient Y_{ij} can be written as

$$Y_{ij} = \sum_{l=0}^{3} \sum_{k=0}^{3} d_{il} x_{lk} d_{kj}, \quad i, j = 0, 1, 2, 3, \tag{9}$$

$$|Y_{ij}| \leq \sum_{l=0}^{3} \sum_{k=0}^{3} |x_{lk}| \cdot |d_{il} d_{kj}| \leq d_{\max} \cdot \sum_{l=0}^{3} \sum_{k=0}^{3} |x_{lk}| = \frac{2}{5} \text{SAD}_{4\times4} \tag{10}$$

for all DCT coefficients. For interblock encoding, the DCT coefficient is quantized as zero when the quantized coefficient Z_{ij} satisfies $|Z_{ij}| < 1$, that is,

$$|Z_{ij}| = \frac{|Y_{ij}|}{Q_{\text{step}}} + \frac{1}{6} < 1. \tag{11}$$

From (10) and (11), it is easy to show that the 4×4 DCT is a zero-block if the sum of absolute differences $\text{SAD}_{4\times4}$ satisfies

$$\text{SAD}_{4\times4} < \frac{25}{12} Q_{\text{step}}. \tag{12}$$

This is Sousa's sufficient condition for zero-block detection.

Moon et al. [7], derived a more precise sufficient condition for zero-block detection by examining the integer

4×4 transform and quantization in H.264/AVC, which is summarized as follows:

(1) if $\mathrm{SAD}_{4\times4} \leq T_0$, then 4×4 DCT is a zero-block, and where

$$T(0) = \frac{\left[5/6 \cdot 2^{15+\lfloor \mathrm{QP}/6 \rfloor}\right]}{\left[4 \cdot M(\mathrm{QP}\%6; 0)\right]};$$ (13)

(2) if $\mathrm{SAD}_{4\times4} > T_0$ and $\mathrm{SAD}_{4\times4} \leq \min\{T_0 + \gamma/2, T_1\}$, then 4×4 DCT is also a zero-block where the parameters $T(1)$ and γ are, respectively, given by

$$T(1) = \frac{\left[5/6 \cdot 2^{15+\lfloor \mathrm{QP}/6 \rfloor}\right]}{\left[2 \cdot M(\mathrm{QP}\%6; 1)\right]},$$

$$\gamma = \min \left\{ \sum_{j=0}^{3} [x_{0j} + x_{3j}], \ \sum_{j=0}^{3} [x_{1j} + x_{2j}] \right\}.$$ (14)

Interestingly, note that $T(0)$ is exactly identical to Sousa's condition. As can be seen, the condition varies with x_{ij}. An intensive study indicates that this sufficient condition varies within a range $2.2Q_{\mathrm{step}} \sim 2.5Q_{\mathrm{step}}$, which is a little higher than the Sousa's condition ($2.08Q_{\mathrm{step}}$).

2.1. A nearly sufficient condition based on ensemble average of DCT coefficients

In this section, a nearly sufficient condition is derived based upon the ensemble average of all 4×4 DCT coefficients by summing up all 4×4 DCT coefficients Y_{ij}. The summation over all DCT coefficients can be written as

$$\sum_{i=0}^{3}\sum_{j=0}^{3} Y_{ij} = C_{00} \cdot x_{00} + C_{01} \cdot x_{01} + C_{02} \cdot x_{02} + \cdots + C_{33} \cdot x_{33}$$

$$= \sum_{i=0}^{3}\sum_{j=0}^{3} C_{ij}x_{ij}.$$ (15)

Define $C_{\max} = \max\{|C_{00}|, |C_{01}|, \ldots, |C_{33}|\}$, and the ensemble average $|Y_{\mathrm{av}}|$ can be upper-bounded as follows:

$$|Y_{\mathrm{av}}| = \frac{1}{16} \left| \sum_{i=0}^{3}\sum_{j=0}^{3} Y_{ij} \right|$$

$$\leq \frac{C_{\max}}{16} \left[|x_{00}| + |x_{01}| + |x_{02}| + \cdots + |x_{33}| \right],$$ (16)

or

$$|Y_{\mathrm{av}}| \leq \frac{C_{\max}}{16} \mathrm{SAD}_{4\times4}.$$ (17)

After some manipulation, $C_{\max}$ was found to be 3.7975. Instead, using $|Y_{ij}|$, if the ensemble average of DCT coefficients $|Y_{\mathrm{av}}|$ is applied to (11), the following upper bound for zero-block detection can be obtained:
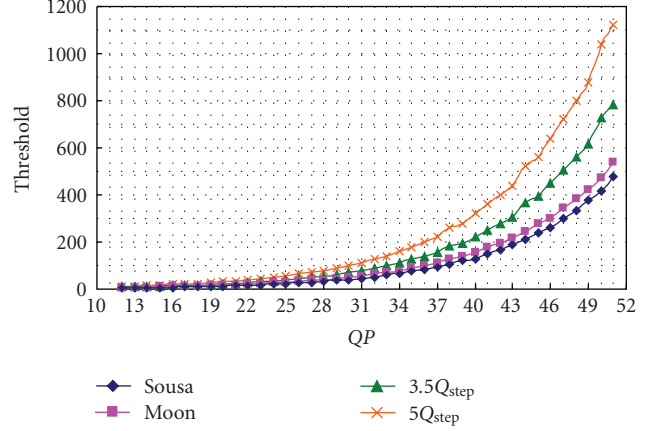
$$\mathrm{SAD}_{4\times4} < 3.5Q_{\mathrm{step}}.$$ (18)

FIGURE 1: Thresholds versus QP.

TABLE 1: Encoding time saving and PSNR loss in DCT/Q.

News	Time Saving (%) (ΔPSNR, dB)			
QP	Sousa	Moon	$3.5Q_{\mathrm{step}}$	$5Q_{\mathrm{step}}$
20	−0.16%	−0.35%	−0.64%(−0.001)	−0.78%(−0.145)
24	−0.56%	−0.84%	−1.09%(0.001)	−1.10%(−0.139)
28	−0.84%	−1.11%	−1.38%(−0.010)	−1.32%(−0.138)
32	−1.19%	−1.47%	−1.78%(0.001)	−1.73%(−0.102)
36	−1.36%	−1.64%	−1.70%(0.002)	−1.89%(−0.069)
40	−1.61%	−1.83%	−1.95%(0.006)	−2.16%(−0.063)
44	−1.81%	−2.04%	−2.36%(−0.026)	−2.87%(−0.046)
48	−2.09%	−2.29%	−3.06%(−0.005)	−4.06%(−0.024)

Although the ensemble average condition is not sufficient and it might detect a zero-block incorrectly, the experiment indicates that only a very small portion of DCT coefficients is incorrectly detected as a zero-block. However, compared to both Sousa's and Moon's conditions, more zero-blocks can be detected correctly using the derived condition.

2.2. Zero-block detection capability and computation reduction in DCT/Quantization

The various thresholds for zero-block detection as a function of QP are plotted in Figure 1. Note that both Sousa's and Moon's conditions are theoretically sufficient, but not for the thresholds $3.5Q_{\mathrm{step}}$ and $5Q_{\mathrm{step}}$. The zero-block detecting capability of all various thresholds carried on the *news* and *paris* sequences are plotted in Figure 2. Although both Sousa's and Moon's conditions are theoretically sufficient, fewer zero-blocks can be detected using these two sufficient conditions compared to the other two conditions. The threshold $5Q_{\mathrm{step}}$ brings out the best zero-block detecting capability; it simultaneously detects numerous improper zero-blocks that could lead to severe performance degradation. The percentage of zero-blocks detected improperly using these two nonsufficient conditions are shown in Figure 3. As can be seen, less than 1% of improper zero-blocks were found for the ensemble average threshold
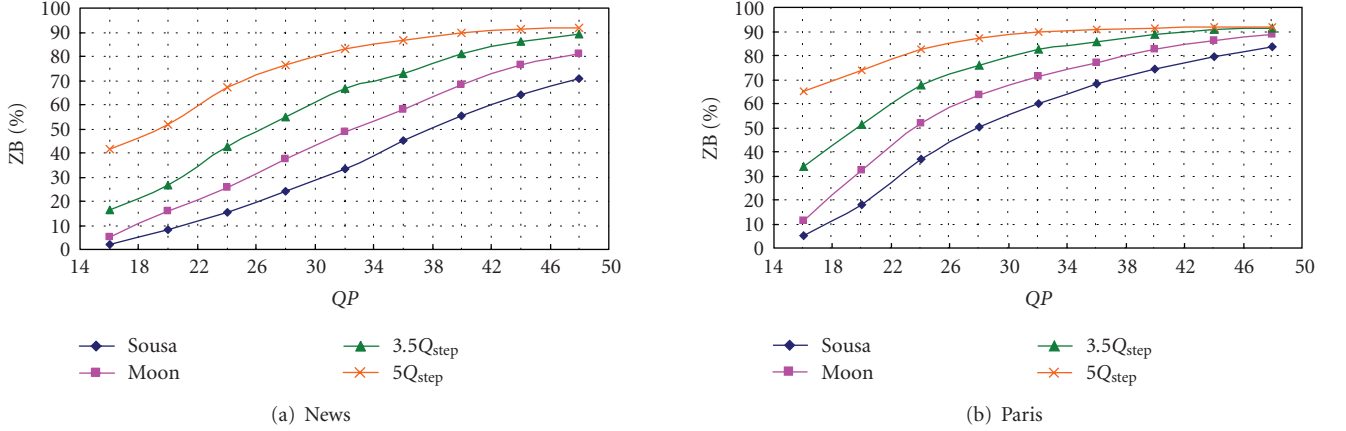
(a) News

(b) Paris

FIGURE 2: Zero-block detecting capability.

$3.5Q_{step}$, while more than 9% for the threshold $5Q_{step}$ for QP = 16.

To evaluate the performance of previously mentioned conditions for early zero-block detection, an experiment was performed in DCT/Q calculation. Table 1 displays the savings of total encoding time in DCT/Q as well as PSNR loss, conducted on the *news* sequence, for different *QPs*. The integer transform and quantization only occupies about 5% of the total encoding time. Note that no loss in either PSNR or bit-rate were found for Sousa's and Moon's conditions. As shown, the threshold $3.5Q_{step}$ can achieve a significant reduction in DCT/Q computation with a negligible PSNR loss. Up to 3% of total encoding time can be saved with PSNR loss of only 0.005 dB for QP = 48. The threshold $5Q_{step}$ [4], however, runs into a severe PSNR degradation due to improper zero-block detection, although computation in DCT/Q can be further reduced. Consequently, the threshold $5Q_{step}$ is not subsequently analyzed.

3. CONVENTIONAL METHODS TO ADOPT ZERO-BLOCK DETECTION IN UMHEXAGON ALGORITHM

In the H.264/AVC, interframe motion estimation is performed for 7 different block sizes (denoted as modes), varying among 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, and 4×4. The motion estimation involves finding a macroblock in a previously encoded reference frame that best matches the current macroblock using SADs between current and reference area samples:

$$
\begin{aligned}
\mathrm{SAD}_{M \times N} &= \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left| C(i,j) - R(i,j) \right| \\
&= \sum_{i=0}^{M/4-1} \sum_{j=0}^{N/4-1} \left\{ \sum_{l=0}^{3} \sum_{k=0}^{3} x_{lk}^{ij} \right\} \\
&= \sum_{i=0}^{M/4-1} \sum_{j=0}^{N/4-1} \mathrm{SAD}_{4 \times 4}^{ij}, \quad M, N = 4, 8 \text{ or } 16.
\end{aligned}
\tag{19}
$$

TABLE 2: Simulation conditions.

Code version	JM86
Profile	Baseline
RDOptimization	on
Entropy coding	CAVLC
Encoding frames	21
Reference frames	5
Frame rate	30
Block type	all
Search range	± 16
Resolution	1/4
GOP structure	IPPP...

TABLE 3: Average search points per frame achieved by various thresholds.

QP	UMH.	U.+Sousa	U.+Moon	U.+$3.5Q_{step}$
20	10.28	10.25	10.15	9.01
24	10.38	10.08	9.76	8.39
28	10.59	9.85	9.51	7.75
32	11.23	10.11	9.75	7.18
36	12.32	10.68	10.06	6.42
40	12.86	10.66	9.49	5
44	13.23	10.05	8.38	3.67
48	14.09	9.11	7.26	3.04

In the early termination method of motion estimation, each $\mathrm{SAD}_{4 \times 4}^{ij}$ in $\mathrm{SAD}_{M \times N}$ is compared with a threshold; and if all $\mathrm{SAD}_{4 \times 4}^{ij}$ satisfy sufficient or nearly sufficient conditions, the motion search stops. In addition, the DCT/Q calculation need not be done if the 4×4 DCT is a zero-block. This leads to a great reduction in computation. Since the conventional early zero-block detection method only requires a comparison of $\mathrm{SAD}_{4 \times 4}^{ij}$ with a threshold, this approach can be applied to all kinds of motion searches, such
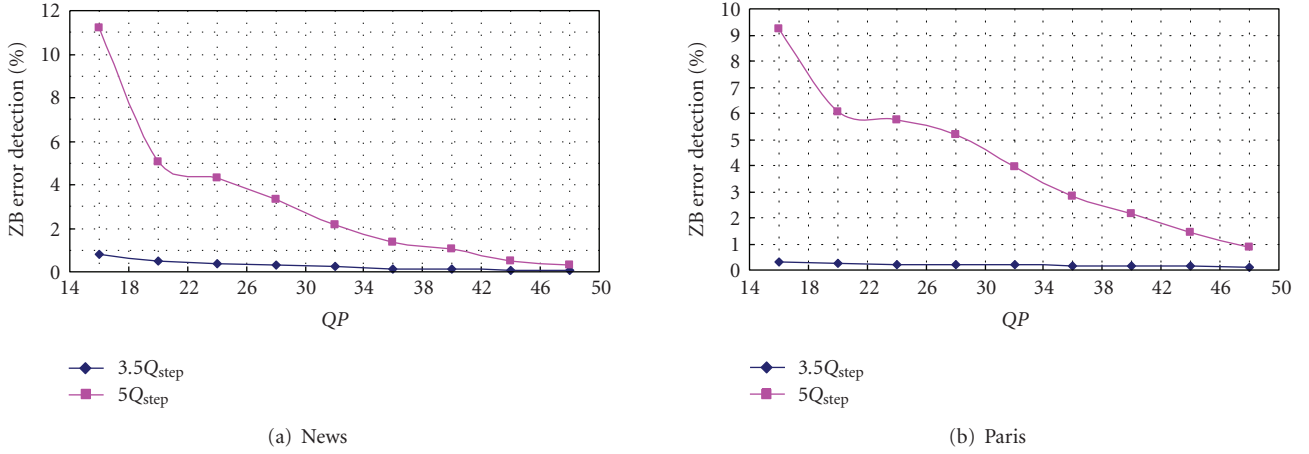
(a) News

(b) Paris

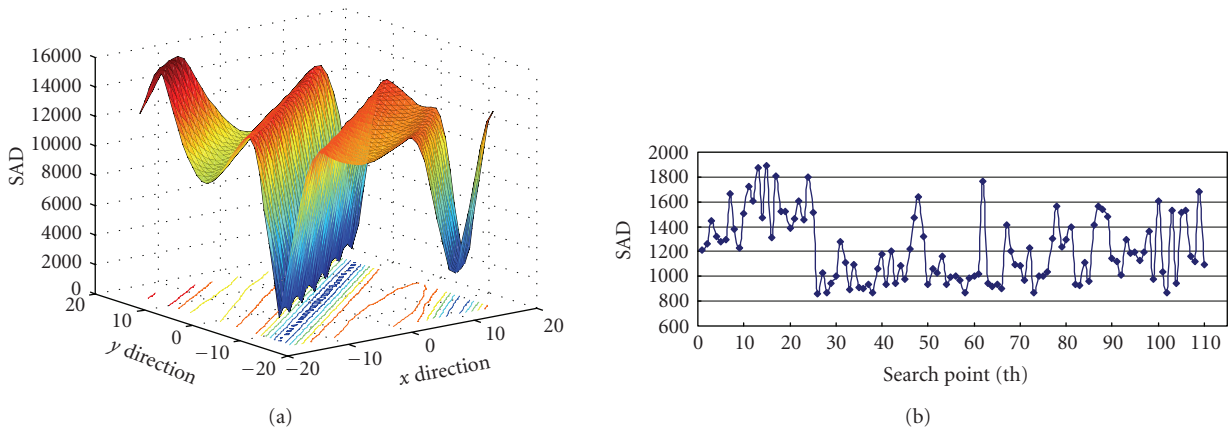FIGURE 3: Percentage of improper zero-block detected.

(a)

(b)

FIGURE 4: Foreman (42nd MB,10th frame) (a) SAD error surface (b) search iteration using UMHexagonS algorithm.

as full search and all other fast search algorithms. This has been investigated in [4, 5].

In this section, we apply the various zero-block detection methods to the UMHexagonS algorithm and investigate the performance. The simulation conditions are tabulated in Table 2. Table 3 displays the average search points per block for different *QPs* conducted on the *news* sequence achieved using various zero-block detection thresholds. As shown, the average search points decrease with increasing threshold. For the *news* sequence and QP = 48, up to 78% of average search points (14.09 reduced to 3.04) in the motion estimation can be saved when utilizing the zero-block detection approach using threshold $3.5Q_{step}$: much higher than the other two sufficient conditions (9.11 and 7.26, resp.). The average PSNR loss, bit-rate increment, and motion estimation time saving versus QP are also compared using various thresholds and tabulated in Table 4. As shown, the early zero-block detection using a nearly sufficient condition (i.e., with threshold $3.5Q_{step}$) significantly outperforms other thresholds in terms of computation for any bit-rate

coding. As high as 56% of motion estimation time can be saved for QP = 48 compared to the UMHexagonS algorithm.

The PSNR degradation, to whatever extent it occurs, becomes strict for low bit-rate coding or high QP. Table 5 displays PSNR loss conducted on several video sequences for QP = 48. As shown, the conventional zero-block detection runs into a PSNR loss of 0.212 dB on the *foreman* sequence. This phenomenon is illustrated in Figure 4, which demonstrates the SAD error surface and the corresponding search iterations using the UMHexagonS algorithm in mode 16×16 for a macroblock (42nd MB, 10th frame) in the *foreman* sequence. As shown, it requires 110 search points for the UMHexagonS algorithm to find the minimum error ($\text{SAD}_{16 \times 16}$ = 864 at the 26th iteration). The search stops at the 26th iteration and the minimum error can also be found when the conventional zero-block detection method is employed to the UMHexagonS algorithm with QP = 30 (threshold $3.5Q_{step}$ = 70). However, the search stops at the first iteration where $\text{SAD}_{16 \times 16}$ = 1210 as QP is increased to QP = 48, which corresponds to the threshold $3.5Q_{step}$ = 560;
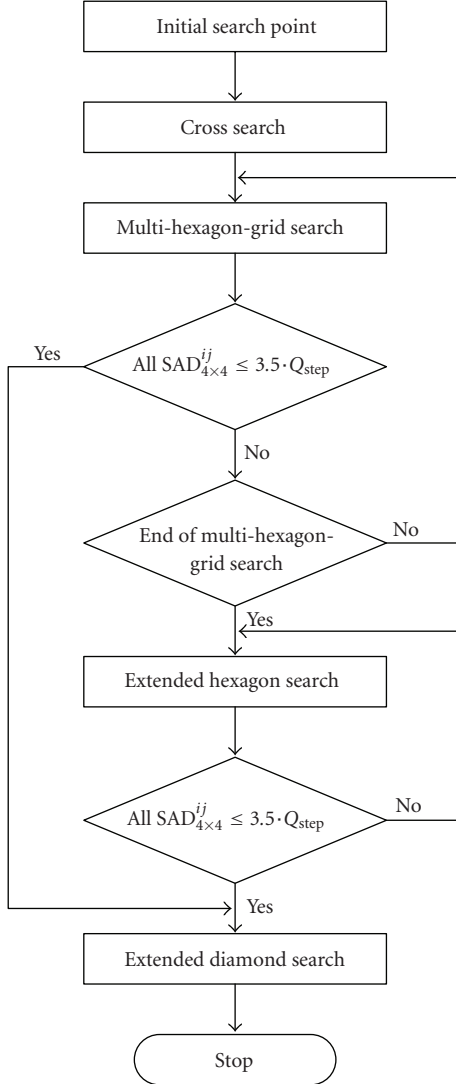
FIGURE 5: Early zero-block detection for motion search and DCT/Q.

TABLE 4: Performance comparison on news sequence.

(a) PSNR loss (dB)

QP	UMH.	U.+Sousa	U.+Moon	U.+3.5Q_{step}
20	43.240	0.000	−0.002	0.004
24	40.786	0.001	−0.001	−0.004
28	38.229	−0.003	−0.002	−0.008
32	35.528	−0.005	−0.009	−0.023
36	32.882	0.002	−0.002	−0.008
40	30.247	0.000	−0.016	−0.046
44	27.596	0.014	−0.017	−0.057
48	25.141	0.000	−0.025	−0.096

(b) Bit-rate (%)

QP	UMH.	U.+Sousa	U.+Moon	U.+3.5Q_{step}
20	619528	0.04%	0.01%	−0.04%
24	375248	0.10%	0.14%	0.29%
28	232072	0.10%	−0.02%	0.05%
32	146304	0.06%	−0.07%	−0.42%
36	92264	−0.29%	0.19%	0.19%
40	58792	−0.38%	−0.39%	0.24%
44	36840	0.46%	0.62%	0.46%
48	22688	0.92%	−0.49%	0.56%

(c) ME time (%)

QP	UMH.	U.+Sousa	U.+Moon	U.+3.5Q_{step}
20	145357	−2.52%	−3.14%	−13.13%
24	142223	−4.41%	−8.75%	−21.10%
28	141105	−9.83%	−13.71%	−27.84%
32	142706	−16.28%	−19.82%	−35.21%
36	139912	−19.19%	−23.62%	−39.93%
40	136736	−21.18%	−26.33%	−45.74%
44	133039	−25.82%	−31.72%	−49.50%
48	129425	−32.14%	−38.22%	−56.62%

TABLE 5: PSNR loss using nearly sufficient condition for QP = 48.

Sequence	UMH.(dB)	UMH. + 3.5Q_{step} (dB)
Mobile	20.164	−0.057
Paris	21.777	−0.06
News	25.141	−0.096
Foreman	23.068	−0.212
Carphone	24.115	−0.129
Claire	27.134	−0.104

and this leads to severe performance degradation. As the quantization parameter increases, the degradation becomes harsher.

4. IMPROVED UMHEXAGONS ALGORITHM

The conventional early zero-block detection technique cannot give a satisfactory coding performance when applied to the UMHexagonS algorithm for large quantization step sizes. In this section, we modify the UMHexagonS algorithm using the early zero-block detection technique to achieve high coding efficiency. Many commonly used video sequences (4 QCIF sequences: *foreman, carphone, football, coastguard* and 4 CIF sequences: *stefan, mobile, paris, tempete*) with different motion contents are simulated by exploiting full search algorithms on these video sequences with a search range $w = \pm16$. The experimental results indicate that a large number of global minimum are occupied near the search center especially at the zero MV (0,0) (average 38%),

horizontal direction (average 27%), and vertical direction (average 18%). The early zero-block detection technique is not employed in these search points to improve coding performance. In addition, the motion search does not stop immediately when the nearly sufficient condition is satisfied. Instead, the diamond search is performed to find a smaller SAD. The improved algorithm is illustrated in Figure 5, and summarized as follows.

TABLE 6: PSNR loss, bit-rate and ME time saving.

| Sequence | QP | PSNR (ΔdB) | | Bit rate (Δ%) | | Time (Δ%) | |
		UMH.	Proposed	UMH.	Proposed	UMH.	Proposed
Mobile	24	35.577	-0.004	2972640	-0.01%	105142	-0.97%
	36	28.051	0.001	410304	0.10%	102171	-7.69%
	48	20.164	$\mathbf{-0.020}$	91088	-0.38%	104937	-33.95%
Paris	24	38.507	0.001	980656	-0.09%	71549	-3.65%
	36	29.662	-0.009	210880	0.13%	73119	-18.76%
	48	21.777	$\mathbf{-0.035}$	41760	0.00%	71174	-43.46%
News	24	40.786	0.005	375248	0.29%	64892	-16.09%
	36	32.882	0.002	92264	0.05%	62842	-35.75%
	48	25.141	$\mathbf{-0.038}$	22688	0.38%	57940	-52.52%
Foreman	24	38.514	-0.011	211368	-0.31%	22155	-1.79%
	36	30.795	-0.013	38560	-1.51%	21732	-20.69%
	48	23.068	$\mathbf{-0.084}$	13368	-1.17%	19830	-54.61%
Carphone	24	40.072	-0.008	199816	0.05%	20701	-7.19%
	36	31.892	0.005	38376	-0.13%	19247	-27.25%
	48	24.115	$\mathbf{-0.038}$	9112	-0.09%	16112	-54.99%
Claire	24	42.565	-0.035	69264	-0.29%	14952	-21.39%
	36	34.724	0.001	15864	-1.36%	14042	-37.47%
	48	27.134	$\mathbf{-0.062}$	5872	-0.77%	12764	-54.36%

Step 1. Predict the initial search point.

Step 2. Perform unsymmetrical-cross search.

Step 3. Perform uneven multi-hexagon-grid search. If all $\mathrm{SAD}_{4\times4}^{ij}$ satisfy the nearly sufficient condition in (16), the motion search stops in this step and jumps to the diamond search in Step 4 and perform the diamond search.

Step 4. Perform extended hexagon based search. Similarly, if all $\mathrm{SAD}_{4\times4}^{ij}$ satisfy the nearly sufficient condition in the hexagon search, then jump to perform the diamond search.

The average PSNR loss, bit-rate increment, and ME time saving of the improved algorithm versus QP are also compared with the UMHexagonS algorithm and tabulated in Table 6. As shown, a great improvement in computation and up to 55% of ME computation can be saved, while maintaining a very good rate distortion performance. A gain of 0.128 dB in PSNR can be obtained for the improved algorithm on the *foreman* sequence for QP = 48 with a slight increase in computation, compared to the conventional early zero-block detection method.

5. CONCLUSION

In this paper, we modified the early termination of UMHexagonS algorithm to avoid the serve performance degradation in high QP. In addition, we derived a nearly sufficient condition for zero-block detection of 4×4 DCT coefficients after quantization, based upon the ensemble average of all 4×4 DCT coefficients. The nearly sufficient

condition for zero-block detection is shown to have excellent zero-block detecting capability, while improper zero-block detection is negligible. The early zero-block detection approach with a nearly sufficient condition (threshold $3.5Q_{\text{step}}$) was then applied to both motion search and DCT/Q calculation in a fast-motion estimation algorithm (UMHexagonS algorithm). The simulation results reveal that a significant improvement in computation reduction (up to 55%) can be achieved with negligible performance degradation compared to the UMHexagonS algorithm.

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[2] Z. Chen, J. Xu, Y. He, and J. Zheng, "Fast integer-pel and fractional-pel motion estimation for H.264/AVC," *Journal of Visual Communication and Image Representation*, vol. 17, no. 2, pp. 264–290, 2006.

[3] Z. Xie, Y. Liu, J. Liu, and T. Yang, "A general method for detecting all-zero-blocks prior to DCT and quantization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 2, pp. 237–241, 2007.

[4] J.-F. Yang, S.-C. Chang, and C.-Y. Chen, "Computation reduction for motion search in low rate video coders," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 10, pp. 948–951, 2002.

[5] L. Yang, K. Yu, J. Li, and S. Li, "An effective variable block-size early termination algorithm for H.264 video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 6, pp. 784–788, 2005.

[6] L. A. Sousa, "General method for eliminating redundant computations in video coding," *Electronics Letters*, vol. 36, no. 4, pp. 306–307, 2000.

[7] Y. H. Moon, G. Y. Kim, and J. H. Kim, "An improved early detection algorithm for all-zero blocks in H.264 video encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 8, pp. 1053–1057, 2005.

[8] D. Wu, K. P. Lim, T. K. Chiew, J. Y. Tham, and K. H. Goh, "An adaptive thresholding technique for the detection of all-zeros blocks in H.264," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '07)*, vol. 5, pp. 329–332, San Antonio, Tex, USA, September 2007.