Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 403634, 12 pages
doi:10.1155/2010/403634

Research Article

A Content-Motion-Aware Motion Estimation for Quality-Stationary Video Coding

Meng-Chun Lin and Lan-Rong Dung

Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan

Correspondence should be addressed to Meng-Chun Lin, asurada.ece90g@nctu.edu.tw

Received 31 March 2010; Revised 3 July 2010; Accepted 1 August 2010

Academic Editor: Mark Liao

The block-matching motion estimation has been aggressively developed for years. Many papers have presented fast block-matching algorithms (FBMAs) for the reduction of computation complexity. Nevertheless, their results, in terms of video quality and bitrate, are rather content-varying. Very few FBMAs can result in stationary or quasistationary video quality for different motion types of video content. Instead of using multiple search algorithms, this paper proposes a quality-stationary motion estimation with a unified search mechanism. This paper presents a content-motion-aware motion estimation for quality-stationary video coding. Under the rate control mechanism, the proposed motion estimation, based on subsample approach, adaptively adjusts the subsample ratio with the motion-level of video sequence to keep the degradation of video quality low. The proposed approach is a companion for all kinds of FBMAs in H.264/AVC. As shown in experimental results, the proposed approach can produce stationary quality. Comparing with the full-search block-matching algorithm, the quality degradation is less than 0.36 dB while the average saving of power consumption is 69.6%. When applying the proposed approach for the fast motion estimation (FME) algorithm in H.264/AVC JM reference software, the proposed approach can save 62.2% of the power consumption while the quality degradation is less than 0.27 dB.

1. Introduction

Motion Estimation (ME) has been proven to be effective to exploit the temporal redundancy of video sequences and, therefore, becomes a key component of multimedia standards, such as MPEG standards and H.26X [1–7]. The most popular algorithm for the VLSI implementation of motion estimation is the block-based full search algorithm [8–11]. The block-based full search algorithm has high degree of modularity and requires low control overhead. However, the full search algorithm notoriously needs high computation load and large memory size [12–14]. The highly computational cost has become a major problem on the implementation of motion estimation.

To reduce the computational complexity of the full-search block-matching (FSBM) algorithm, researchers have proposed various fast algorithms. They either reduce search steps [12, 15–22] or simplify calculations of error criterion [8, 23–25]. Some researchers combined both step-reduction and criterion-simplifying to significantly reduce computational load with little degradation. By combining step-reduction and criterion-simplifying, some researchers proposed two-phase algorithms to balance the performance between complexity and quality [26–28]. These fast algorithms have been shown that they can significantly reduce the computational load while the average quality degradation is little. However, a real video sequence may have different types of content, such as slow-motion, moderate-motion, and fast-motion, and little quality degradation in average does not imply the quality is acceptable all the time. The fast block-matching algorithms (FBMAs) mentioned above are all independent of the motion type of video content, and their quality degradation may considerably vary within a real video sequence.

Few papers present quality-stationary motion estimation algorithms for video sequences with mixed fast-motion, moderate-motion, and slow-motion content. Huang et al. [29] propose an adaptive, multiple-search-pattern FBMA, called the A-TDB algorithm, to solve the content-dependent problem. Motivated by the characteristics of three-step search (TSS), diamond search (DS), and block-based gradient descent search (BBGDS), the A-TDB algorithm dynamically switches search patterns according to the motion type of video content. Ng et al. [30] propose an adaptive search patterns switching (SPS) algorithm by using an efficient motion content classifier based on error descent rate (EDR) to reduce the complexity of the classification process of the A-TDB algorithm. Other multiple search algorithms have been proposed [31, 32]. They showed that using multiple search patterns in ME can outperform stand-alone ME techniques.

Instead of using multiple search algorithms, this paper intends to propose a quality-stationary motion estimation with a unified search mechanism. The quality-stationary motion estimation can appropriately adjust the computational load to deliver stationary video quality for a given bitrate. Herein, we used the subsample or pixel-decimation approach for the motion-vector (MV) search. The use of subsample approach is two-folded. First, the subsample approach can be applied for all kinds of FBMAs and provide high degree of flexibility for adaptively adjusting the computational load. Secondly, the subsample approach is feasible and scalable for either hardware or software implementation. The proposed approach is not limited for FSBM, but valid for all kinds of FBMAs. The proposed approach is a companion for all kinds of FBMAs in H.264/AVC.

Articles in [33–38] present the subsample approaches for motion estimation. The subsample approaches are used to reduce the computational cost of the block-matching criterion evaluation. Because the subsample approaches always desolate some pixels, the accuracy of the estimated MVs becomes the key issue to be solved. As per the fundamental of sampling, downsampling a signal may result in aliasing problem. The narrower the bandwidth of the signal, the lower the sampling frequency without aliasing problem will be. The published papers [33–38] mainly focus on the subsample pattern based on the intraframe high-frequency pixels (i.e., edges). Instead of considering spatial frequency bandwidth, to be aware of the content motion, we determine the subsample ratio by temporal bandwidth. Applying high subsample ratio for slow motion blocks would not reduce the accuracy for slow motion or result in large amount of prediction residual. Note that the amount of prediction residual is a good measure of the compressibility. Under a fixed bit-rate constraint, the compressibility affects the compression quality. Our algorithm can adaptively adjust the subsample ratio with the motion-level of video sequence. When the interframe variation becomes high, we consider the motion-level of interframe as the fast-motion and apply low subsample ratio for motion estimation. When the interframe variation becomes low, we apply high subsample ratio for motion estimation.

Given the acceptable quality in terms of PSNR and bitrate, we successfully develop an adaptive motion estimation algorithm with variable subsample ratios. The proposed algorithm is awared of the motion-level of content and adaptively select the subsample ratio for each group of picture (GOP). Figure 1 shows the application of proposed algorithm. The scalable fast ME is an adjustable motion estimation whose subsampling ratio can be tuned by the motion-level detection. The dash-lined region is the proposed motion estimation algorithm and the proposed algorithm switches the subsample ratios according to the zero motion vector count (ZMVC). The higher the ZMVC, the higher the subsample ratio. As the result of applying the algorithm for H.264/AVC applications, the proposed algorithm can produce stationary quality at the PSNR of 0.36 dB for a given bitrate while saving about 69.6% power consumption for FSBM, and the PSNR of 0.27 dB and 62.2% power-saving for FBMA. The rest of the paper is organized as follows. In Section 2, we introduce the generic subsample algorithm in detail. Section 3 describes the high-frequency aliasing problem in the subsample algorithm. Section 4 describes the proposed algorithm. Section 5 shows the experimental performance of the proposed algorithm in H.264 software model. Finally, Section 6 concludes our contribution and merits of this work.

2. Generic Subsample Algorithm

Among many efficient motion estimation algorithms, the FSBM algorithm with sum of absolute difference (SAD) is the most popular approach for motion estimation because of its considerably good quality. It is particularly attractive to ones who require extremely high quality, however, it requires a huge number of arithmetic operations and results in highly computational load and power dissipation. To efficiently reduce the computational complexity of FSBM, lots of published papers have efficiently presented fast algorithms for motion estimation. For these fast algorithms, much research addresses subsample technologies to reduce the computational load of FSBM [33–37, 39, 40]. Liu and Zaccarin [33], as pioneers of subsample algorithm, applied subsampling technology to FSBM and significantly reduced the computation load. Cheung and Po [34] well proposed a subsample algorithm combined with hierarchical-search method. Here, we present a generic subsample algorithm in which the subsample ratio ranges from 16-to-2 to 16-to-16. The basic operation of the generic subsample algorithm is to find the best motion estimation with less SAD computation. The generic subsample algorithm uses (1) as a matching criterion, called the subsample sum of absolute difference (SSAD), where the macroblock size is N-by-N, $R(i,j)$ is the luminance value at (i, j) of the current macroblock (CMB). The $S(i + u, j + v)$ is the luminance value at (i, j) of the reference macroblock (RMB) which offsets (u, v) from the CMB in the searching area $2p$-by-$2p$. $\mathrm{SM}_{16:2m}$ is the subsample mask for the subsample ratio 16-to-$2m$ as shown in (2) and the subsample mask $\mathrm{SM}_{16:2m}$ is generated from basic mask (BM) as shown in (3), When the subsample ratios are fixed at powers of two because of regularly spatial distribution, these ratios are 16 : 16, 16 : 8, 16 : 4, and 16 : 2, respectively. These subsample masks can be generated in a 16-by-16 macroblock by using (3) and are shown in Figure 2. From (3), given a subsample mask generated, the computational cost of SSAD can be lower than that of

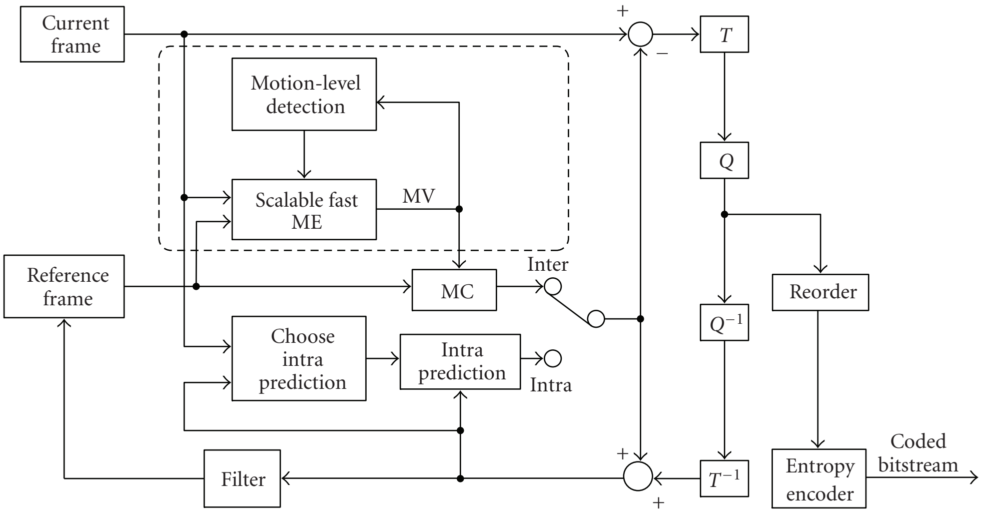


Figure 1: The proposed system diagram for H.264/AVC encoder.

SAD calculation, hence, the generic subsample algorithm can achieve the goal of power-saving with flexibly changing subsample ratio. However, the generic subsample algorithm suffers aliasing problem for high-frequency band. The aliasing problem will degrade the validity of motion vector (MV) and obviously result in a visual quality degradation for some video sequences. The next section will describe how the high-frequency aliasing problem occurs for subsample algorithm in detail,

$$\begin{aligned}&\mathrm{SSAD}_{\mathrm{SM}_{16:2m}}(u,v)\\&\quad=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|\mathrm{SM}_{16:2m}(i,j)\cdot\left(S(i+u,j+v)-R(i,j)\right)\right|,\\&\qquad\text{for } -p\le u,\ v\le p-1,\end{aligned}\tag{1}$$

$$\begin{aligned}&\mathrm{SM}_{16:2m}(i,j)=\mathrm{BM}_{16:2m}(i \bmod 4, j \bmod 4)\\&\qquad\text{for } m=1,2,3,4,5,6,7,8,\end{aligned}\tag{2}$$

$$\begin{aligned}&\mathrm{BM}_{16:2m}(k,l)\\&\quad=\begin{bmatrix}u(m-1)&u(m-5)&u(m-2)&u(m-6)\\u(m-7)&u(m-3)&u(m-8)&u(m-4)\\u(m-2)&u(m-5)&u(m-1)&u(m-6)\\u(m-7)&u(m-3)&u(m-8)&u(m-4)\end{bmatrix}\\&\qquad\text{for } 0\le k,\ l\le 3,\end{aligned}\tag{3}$$

where $u(n)$ is a step function; that is,

$$u(n)=\begin{cases}1, & \text{for } n\ge 0,\\0, & \text{for } n<0.\end{cases}\tag{4}$$

3. High-Frequency Aliasing Problem

According to sampling theory [41], the decrease of sampling frequency will result in aliasing problem for high-frequency band. On the other hand, when the bandwidth of signal is narrow, higher downsample ratio or lower sampling frequency is allowed without aliasing problem. When applying the generic subsample algorithm for video compression, for high-variation sequences, the aliasing problem occurs and leads to considerable quality degradation because the high-frequency band is messed up. Papers [42, 43] hence propose adaptive subsample algorithms to solve the problem. They employed the variable subsample pattern for spatial high-frequency band, that is, edge pixels. However, the motion estimation is used for interframe prediction and temporal high-frequency band should be mainly treated carefully. Therefore, we determine the subsample ratio by the interframe variation. The interframe variation can be characterized by the motion-level of content. The ZMVC is a good sign for the motion-level detection because it is feasible for measurement and requires low computation load. The high ZMVC means that the interframe variation is low and vice versa. Hence, we can set high subsample ratio for high ZMVCs and low subsample ratio for low ZMVCs. Doing so, the aliasing problem can be alleviated and the quality can be frozen within an acceptable range.

To start with, we first analyze the results of visual quality degradation with different subsample ratios. We simulated the moderate motion video sequence "table" in H.264 JM10.2 software, where the length of GOP is fifteen frames, the frame rate is 30 frames/s, the bit rate is 450 k bits/s, and initial Q_p is 34. After applying three subsample ratios of 16 : 8, 16 : 4, and 16 : 2, Figure 3 shows quality degradation results versus subsample ratios. The average quality degradation of the ith GOP ($\Delta Q_{i\text{th}\,GOP}$) is defined as (5), where $\mathrm{PSNRY}_{i\,\mathrm{FSBM}}$ is the average PSNRY of ith GOP using the full-search block-matching (FSBM) and $\mathrm{PSNRY}_{i\,\mathrm{SSR}}$ is the average PSNRY of ith GOP with specific subsample ratio (SSR). From Figure 3, although the video sequence "table" is, in the literature, regarded as a moderate motion, there exists the high interframe variation between the third GOP and the seventh GOP. Obviously,

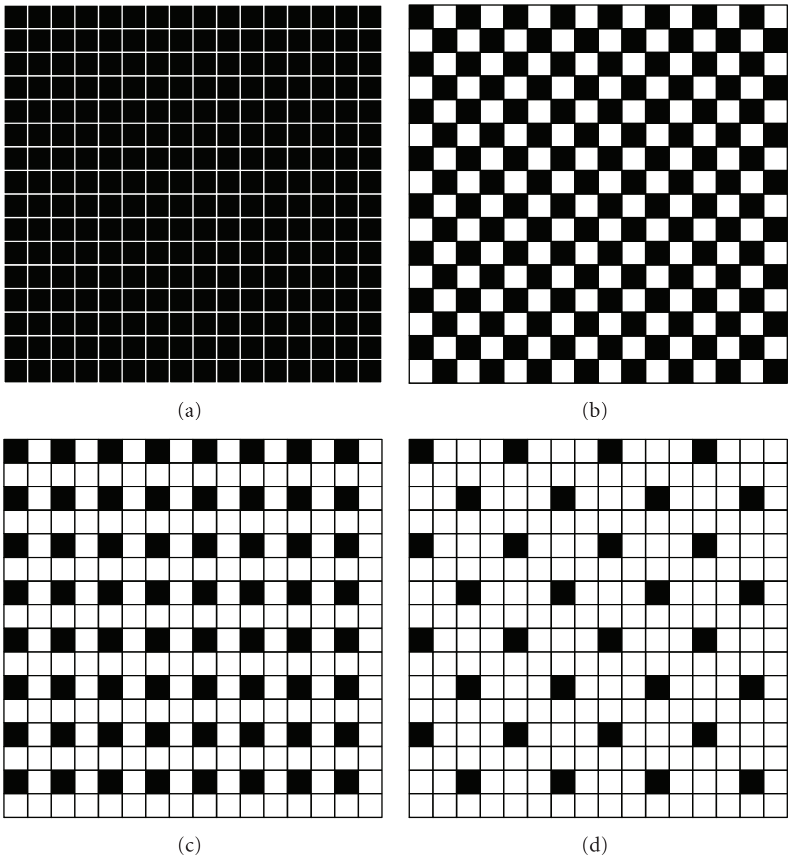


Figure 2: (a) 16 : 16 subsample pattern, (b) 16 : 8 subsample pattern, (c) 16 : 4 subsample pattern and (d) 16 : 2 subsample pattern.

applying the higher subsample ratios may result in serious aliasing problem and higher degree of quality degradation. In contrast, between the eleventh GOP and the twentieth GOP, the quality degradation is low for lower subsample ratios. Therefore, we can vary the subsample ratio with the motion-level of content to produce quality-stationary video while saving the power consumption when necessary. Accordingly, we developed a content-motion-aware motion estimation based on the motion-level detection. The proposed motion estimation is not limited for FSBM, but valid for all kinds of FBMAs,

$$\Delta Q_{i\text{th GOP}} = (\text{PSNRY}_{i\,\text{FSBM}} - \text{PSNRY}_{i\,\text{SSR}}). \tag{5}$$

4. Adaptive Motion Estimation with Variable Subsample Ratios

To efficiently alleviate the high-frequency aliasing problem and maintain the visual quality for video sequences with variable motion levels, we propose an adaptive motion estimation algorithm with variable subsample ratios, called the Variable Subsampling Motion Estimation (VSME). The proposed algorithm determines the suitable subsample ratio for each GOP based on the ZMVC. The algorithm can be applied for FSBM algorithm and all other FBMAs. The ZMVC is a feasible measurement for indicating the motion-level of video. The higher the ZMVC, the lower the motion-level. Figure 4 shows the ZMVC of first P-frame in each GOP for table sequence. From Figures 3 and 4, we can see that when the ZMVC is high the ΔQ for the subsample ratio of 16 : 2 is little. Since the tenth GOP is the scene-changing segment, all subsampling algorithms will fail to maintain the quality. Between the third and seventh GOPs, ΔQ becomes high and the ZMVC is relatively low. Thus, this paper uses the ZMVC as a reference to determine the suitable subsample ratio.

In the proposed algorithm, we determine the subsample ratio at the beginning of each GOP because the ZMVC of the first interframe prediction is the most accurate. The reference frame in the first interframe prediction is a reconstructed I-frame but others are not for each GOP. Only the reconstructed I-frame does not incur the influence resulted from the quality degradation of the inaccurate interframe prediction. That is, we only calculate the ZMVC of the first P-frame for the subsample ratio selection to efficiently save the computational load of ZMVC. Note that the ZMVC of the first P-frame is calculated by using 16 : 16 subsample ratio. Given the ZMVC of the first P-frame, the motion-level is determined by comparing the ZMVC with preestimated threshold values. The threshold values is decided statistically using popular video clips.

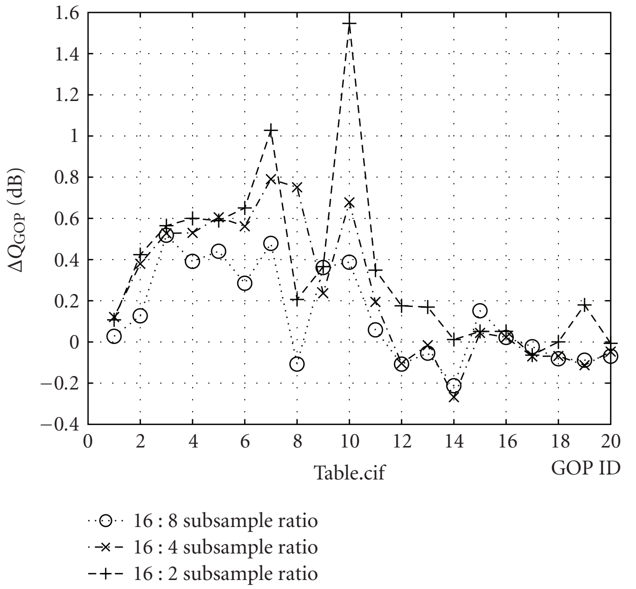


Figure 3: The diagram of ΔQ with 16 : 8, 16 : 4, 16 : 2 subsample ratios for table sequence.

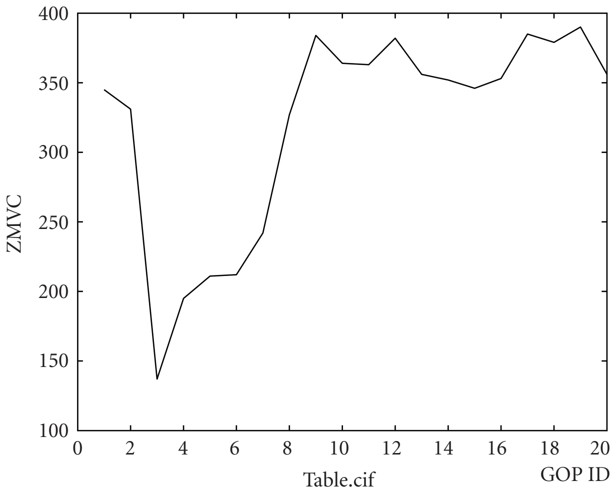


Figure 4: The ZMVC of each GOP for table sequence.

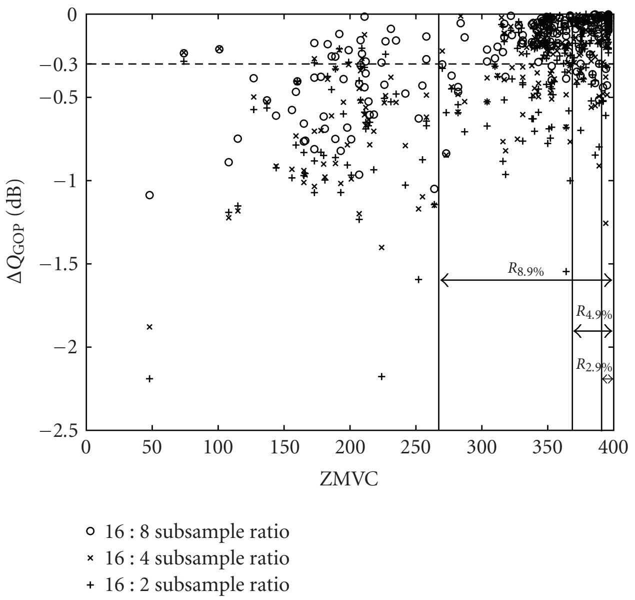


Figure 5: The statistical distribution of ΔGOP versus ZMVC.

Table 1: Threshold setting for different conditions under the 0.3 *dB* of visual quality degradation.

	$p = 90$	$p = 85$	$p = 80$	$p = 75$	$p = 70$	$p = 65$
$k = 2$	393	387	376	344	305	232
$k = 4$	368	356	344	251	239	190
$k = 8$	265	242	227	297	179	49

Table 2: Testing video sequences.

	Video sequence	Number of frames
Fast Motion	Dancer	250
	Foreman	300
	Flower	250
Normal Motion	Table	300
	Mother_Daughter (M_D)	300
	Weather	300
	Children	300
	Paris	300
Slow Motion	News	300
	Akiyo	300
	Silent	300
	Container	300

To set the threshold values for motion-level detection, we first built up the statistical distribution of ΔQ versus ZMVC for video sequences with subsample ratios of 16 : 2, 16 : 4, 16 : 8, and 16 : 16. Figure 5 illustrates the distribution. Then, we calculated the coverage of given PSNR degradation ΔQ. In the video coding community, 0.5 dB is empirically considered a threshold below which the perceptual quality difference cannot be perceived by subjects. The quality degradation of greater than 0.5 dB is sensible for human perception [44]. To keep the degradation of video quality low for the quality-stationary video coding, a strict threshold of smaller than 0.5 dB is assigned to be a aimed ΔQ without the sensible quality degradation. Therefore, in this paper, the aimed ΔQ is 0.3 dB. We use the coverage range $R_{k,p\%}$ to set the threshold values for motion-level detection. The motion-level detection will further determine the subsample ratio. The range $R_{k,p\%}$ indicates the covered range of ZMVC, where $p\%$ is the percentage of GOPs whose ΔQ is less than 0.3 dB for subsample ratio of 16 : k. Given the parameters p and k, we can set threshold values as shown in Table 1.

Table 3: Analysis of quality degradation using three adaptive subsample rate decisions.

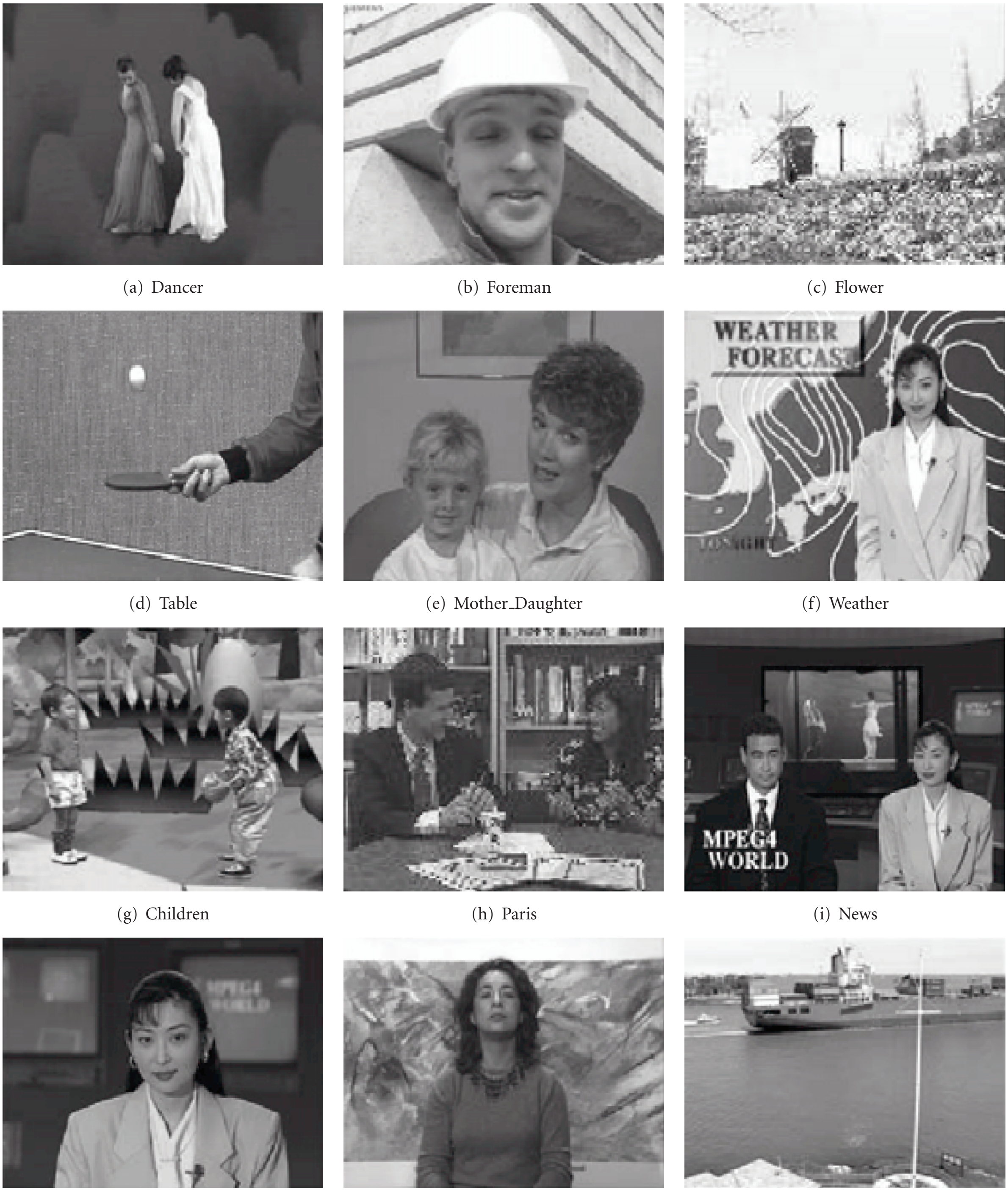
	$p = 90$	$p = 85$	$p = 80$	$p = 75$	$p = 70$	$p = 65$
Dancer	−0.02	−0.02	−0.02	−0.09	−0.36	−0.77
Foreman	−0.09	−0.15	−0.16	−0.31	−0.33	−0.59
Flower	0	−0.04	−0.04	−0.15	−0.27	−0.44
Table	−0.05	−0.06	−0.11	−0.19	−0.26	−0.34
M_D	−0.2	−0.22	−0.23	−0.33	−0.36	−0.45
Weather	−0.2	−0.22	−0.25	−0.29	−0.33	−0.33
Children	−0.13	−0.16	−0.19	−0.28	−0.29	−0.29
Paris	−0.17	−0.22	−0.21	−0.31	−0.35	−0.35
News	−0.08	−0.1	−0.12	−0.15	−0.2	−0.20
Akiyo	−0.09	−0.12	−0.12	−0.15	−0.15	−0.15
Silent	−0.06	−0.05	−0.04	−0.06	−0.09	−0.09
Container	−0.02	−0.02	−0.02	−0.02	−0.02	−0.02

Table 4: Analysis of average subsample ratio using three adaptive subsample rate decisions.

	$p = 90$	$p = 85$	$p = 80$	$p = 75$	$p = 70$
Dancer	16 : 15.55	16 : 15.55	16 : 15.55	16 : 14.43	16 : 11.75
Foreman	16 : 14.32	16 : 13.31	16 : 12.93	16 : 10.61	16 : 10.24
Flower	16 : 16.00	16 : 15.10	16 : 15.10	16 : 11.98	16 : 8.80
Table	16 : 9.50	16 : 9.03	16 : 7.17	16 : 5.32	16 : 4.57
M_D	16 : 7.08	16 : 6.43	16 : 6.34	16 : 3.92	16 : 3.55
Weather	16 : 5.87	16 : 5.32	16 : 4.39	16 : 3.18	16 : 3.00
Children	16 : 7.82	16 : 7.27	16 : 6.43	16 : 3.83	16 : 3.27
Paris	16 : 6.52	16 : 6.25	16 : 5.22	16 : 3.46	16 : 3.00
News	16 : 7.45	16 : 6.71	16 : 4.95	16 : 3.09	16 : 3.00
Akiyo	16 : 4.76	16 : 3.83	16 : 3.46	16 : 3.00	16 : 3.00
Silent	16 : 7.27	16 : 7.08	16 : 6.34	16 : 3.92	16 : 3.00
Container	16 : 3.18	16 : 3.00	16 : 3.00	16 : 3.00	16 : 3.00
Average	16 : 8.58	16 : 8.04	16 : 7.35	16 : 5.60	16 : 4.87

Table 5: Performance analysis of quality degradation for various video sequences using various methods. (Note that the proposed algorithm can always keep the quality degradation low.)

Full search block matching (FSBM) algorithm									
	Generic	Generic	Generic	Generic	Generic	Generic	Generic	Generic	Proposed
Video	16 : 16	16 : 14	16 : 12	16 : 10	16 : 8	16 : 6	16 : 4	16 : 2	algorithm
sequence	subsample	subsample	subsample	subsample	subsample	subsample	subsample	subsample	(70%)
	ratio	ratio	ratio	ratio	ratio	ratio	ratio	ratio	
	PSNRY	ΔPSNRY	ΔPSNRY	ΔPSNRY	ΔPSNRY	ΔPSNRY	ΔPSNRY	ΔPSNRY	ΔPSNRY
Dancer	34.42	−0.18	−0.33	−0.53	−0.7	−0.86	−0.92	−0.93	−0.36
Foreman	30.51	−0.09	−0.18	−0.27	−0.4	−0.55	−0.72	−0.78	−0.33
Flower	20.58	−0.05	−0.1	−0.18	−0.28	−0.4	−0.49	−0.51	−0.27
Table	32.04	−0.02	−0.04	−0.09	−0.13	−0.16	−0.24	−0.35	−0.26
M_D	40.34	−0.03	−0.02	−0.08	−0.15	−0.25	−0.35	−0.46	−0.36
Weather	33.26	−0.06	−0.1	−0.09	−0.15	−0.22	−0.28	−0.33	−0.33
Children	30	−0.01	−0.05	−0.11	−0.14	−0.17	−0.22	−0.29	−0.29
Paris	31.67	0	−0.04	−0.05	−0.1	−0.13	−0.27	−0.33	−0.35
News	38.27	−0.02	−0.01	−0.04	−0.06	−0.09	−0.13	−0.22	−0.2
Akiyo	43.36	0.01	−0.01	−0.02	−0.03	−0.05	−0.09	−0.16	−0.15
Silent	35.62	−0.03	−0.03	−0.03	−0.02	−0.02	−0.06	−0.08	−0.09
Container	36.47	0	−0.01	−0.01	0	−0.02	−0.02	−0.02	−0.02



(a) Dancer (b) Foreman (c) Flower

(d) Table (e) Mother_Daughter (f) Weather

(g) Children (h) Paris (i) News

(j) Akiyo (k) Silent (l) Container

Figure 6: Test clips: (a) Dancer, (b) Foreman, (c) Flower, (d) Table, (e) Mother_Daughter, (M_D) (f) Weather, (g) Children, (h) Paris, (i) News, (j) Akiyo, (k), and Silent (l) Container.

5. Selection of ZMVC Threshold and Simulation Results

The proposed algorithm is simulated for H.264 video coding standard by using software model JM10.2 [45]. Here, we use twelve famous video sequences [46] to simulate in JM10.2, and they are shown in Figure 6 and Table 2. From Table 2, the file format of these video sequences is CIF (352×288 pixels) and the search range is ±16 in both horizontal and vertical directions for a 16-16 macroblock. The bit-rate control fixes the bit rate of 450 k under displaying 30 frames/s. The selection of threshold values is based on two factors: average

Table 6: Performance analysis of speedup ratio.

Full search block matching (FSBM) algorithm									
Video sequence	Generic 16 : 16 subsample ratio	Generic 16 : 14 subsample ratio	Generic 16 : 12 subsample ratio	Generic 16 : 10 subsample ratio	Generic 16 : 8 subsample ratio	Generic 16 : 6 subsample ratio	Generic 16 : 4 subsample ratio	Generic 16 : 2 subsample ratio	Proposed algorithm (70%) ratio
	Speedup	Speedup	Speedup	Speedup	Speedup	Speedup	Speedup	Speedup	Speedup
Dancer	1	1.143	1.3334	1.60011	2.0001	2.6671	4.0006	8.0012	1.36
Foreman	1	1.143	1.3334	1.60013	2.0002	2.6669	4.0003	8.0006	1.56
Flower	1	1.143	1.3334	1.60011	2.0001	2.6671	4.0006	8.0012	1.82
Table	1	1.143	1.3334	1.60013	2.0002	2.6669	4.0003	8.0006	3.50
M_D	1	1.143	1.3334	1.60013	2.0002	2.6669	4.0003	8.0006	4.50
Weather	1	1.143	1.3334	1.60013	2.0002	2.6669	4.0003	8.0006	5.33
Children	1	1.143	1.3334	1.60013	2.0002	2.6669	4.0003	8.0006	4.89
Paris	1	1.143	1.3334	1.60013	2.0002	2.6669	4.0003	8.0006	5.33
News	1	1.143	1.3334	1.60013	2.0002	2.6669	4.0003	8.0006	5.33
Akiyo	1	1.143	1.3334	1.60013	2.0002	2.6669	4.0003	8.0006	5.33
Silent	1	1.143	1.3334	1.60013	2.0002	2.6669	4.0003	8.0006	5.33
Container	1	1.143	1.3334	1.60013	2.0002	2.6669	4.0003	8.0006	5.33

Table 7: Performance analysis of quality degradation for various video sequences using various methods. (Note that the proposed algorithm can always keep the quality degradation low.)

Fast motion estimation (FME) algorithm									
Video sequence	Generic 16 : 16 subsample ratio	Generic 16 : 14 subsample ratio	Generic 16 : 12 subsample ratio	Generic 16 : 10 subsample ratio	Generic 16 : 8 subsample ratio	Generic 16 : 6 subsample ratio	Generic 16 : 4 subsample ratio	Generic 16 : 2 subsample ratio	Proposed algorithm (70%)
	PSNRY	ΔPSNRY	ΔPSNRY	ΔPSNRY	ΔPSNRY	ΔPSNRY	ΔPSNRY	ΔPSNRY	ΔPSNRY
Dancer	33.48	−0.17	−0.31	−0.47	−0.63	−0.84	−1.01	−0.99	−0.05
Foreman	29.63	−0.06	−0.11	−0.17	−0.21	−0.29	−0.45	−0.69	−0.08
Flower	19.64	−0.01	−0.03	−0.06	−0.08	−0.15	−0.25	−0.48	−0.01
Table	31.07	−0.02	−0.03	−0.06	−0.07	−0.11	−0.17	−0.25	−0.09
M_D	39.44	0	0	−0.02	−0.02	−0.05	−0.12	−0.31	−0.24
Weather	32.34	−0.01	−0.02	−0.05	−0.09	−0.07	−0.13	−0.27	−0.26
Children	29.12	−0.06	−0.08	−0.02	−0.15	−0.16	−0.23	−0.3	−0.27
Paris	30.69	0.04	0.02	0.04	0.04	0.01	−0.05	−0.21	−0.15
News	37.29	0.03	0.05	0.03	0.05	0.05	0.03	−0.05	−0.05
Akiyo	42.38	0.03	0.04	0.03	0.02	−0.01	−0.02	−0.07	−0.08
Silent	34.64	0	0	0	0	0.04	0.05	0.02	0
Container	35.5	0	0.02	0.01	0	0	0.01	−0.03	−0.02

quality degradation (Δ PSNRY) and average subsample ratio. The PSNRY is defined as

$$\mathrm{PSNRY} = 10\log\frac{255^2}{(1/NM)\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\left[I_Y(x,y)-\hat{I}_Y(x,y)\right]^2}, \tag{6}$$

where the frame size is $N \times M$, and $I_Y(x,y)$ and $\hat{I}_Y(x,y)$ denote the Y components of original frame and reconstructed frame at (x,y). The quality degradation ΔPSNRY is the PSNRY difference between the proposed algorithm and FSBM algorithm with 16-to-16 subsample ratio.

The average subsample ratio is another index for subsample ratio selection, as defined in (7) where $N_P(k)$ are the P-frames subsampled by 16 : k. Later, we will use it to estimate the average power consumption of the proposed algorithm,

$$\text{Average subsample ratio} = 16 : \frac{N_P(16)*16+N_P(8)*8+N_P(4)*4+N_P(2)*2}{\text{number of P-frames}} \tag{7}$$

Table 3 shows the simulation results of ΔPSNRY for these tested video sequences with different set of threshold values.

Table 8: Performance analysis of speedup ratio.

	Fast motion estimation (FME) algorithm								
Video sequence	Generic 16 : 16 subsample ratio	Generic 16 : 14 subsample ratio	Generic 16 : 12 subsample ratio	Generic 16 : 10 subsample ratio	Generic 16 : 8 subsample ratio	Generic 16 : 6 subsample ratio	Generic 16 : 4 subsample ratio	Generic 16 : 2 subsample ratio	Proposed Algorithm (70%) ratio
	Speedup	Speedup	Speedup	Speedup	Speedup	Speedup	Speedup	Speedup	Speedup
Dancer	1	1.147252	1.346325	1.626553	2.051174	2.768337	4.208802	8.5202	1.056017
Foreman	1	1.14796	1.34685	1.6294	2.05981	2.78782	4.25265	8.2275502	1.16797
Flower	1	1.143542	1.335488	1.603778	2.006855	2.63666	3.975399	8.096571	1.061454
Table	1	1.150301	1.352315	1.637259	2.067149	2.7824	4.210231	8.497531	2.50664
M_D	1	1.150295	1.349931	1.627438	2.040879	2.724727	4.086932	8.16456	4.611836
Weather	1	1.153651	1.36162	1.653674	2.092012	2.815901	4.250473	8.529343	5.379942
Children	1	1.219562	1.488654	1.719515	2.569355	3.51429	5.697292	12.43916	5.056478
Paris	1	1.15079	1.354444	1.645437	2.083324	2.812825	4.270938	8.627448	5.422681
News	1	1.150716	1.351302	1.631096	2.04845	2.740255	4.12047	8.253857	5.260884
Akiyo	1	1.145874	1.340152	1.61157	2.017577	2.692641	4.04448	8.080182	5.35473
Silent	1	1.15267	1.355195	1.63785	2.060897	2.7634	4.160839	8.338212	4.845362
Container	1	1.149457	1.348652	1.626109	2.0412	2.731408	4.109702	8.226404	5.428775

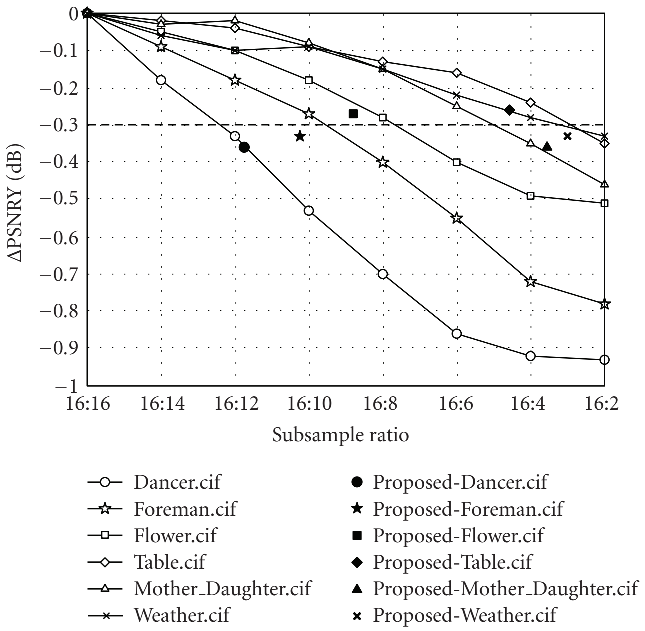


Figure 7: The quality degradation chart of FSBM with fixed subsample ratios and proposed algorithm.

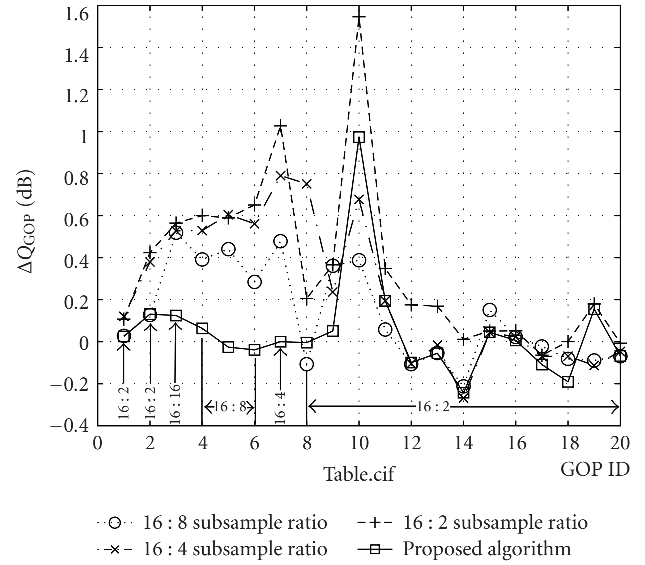


Figure 8: The dynamic quality degradation of the clip "Table" with fixed subsample ratios and proposed algorithm.

From Table 3, the set of threshold values with $p \geq 80$ can satisfy all tested video sequences under the average quality degradation of 0.3 dB; however, the overall average subsample ratios shown in Table 4 are lower than the others. The lower the subsample ratio, the higher the computational power will be. The uses of the set of threshold values of $p = 70$ and $p = 75$ also result in the quality degradations less than 0.36 dB which is close to the 0.3 dB goal. To achieve the goal of the quality degradation under the low computational power, the set of threshold values with $p = 70$ is favored in this paper. As shown in Table 4, the use of the set of threshold values of $p = 70$ results in the quality degradations less than 0.36 dB which is close to the 0.3 dB goal while the power consumption reduction is 69.6% comparing with FSBM without downsampling.

After choosing the set of threshold values between 16 : 16, 16 : 8, 16 : 4, and 16 : 2, we compare the proposed algorithm with generic subsample rate algorithms. Table 5 illustrates the simulation results. Figure 7 illustrates the distribution diagram of ΔPSNRY versus subsample ratio based on Table 5. From Figure 7, to maintain ΔPSNRY around 0.3 dB, the generic algorithm must at least use

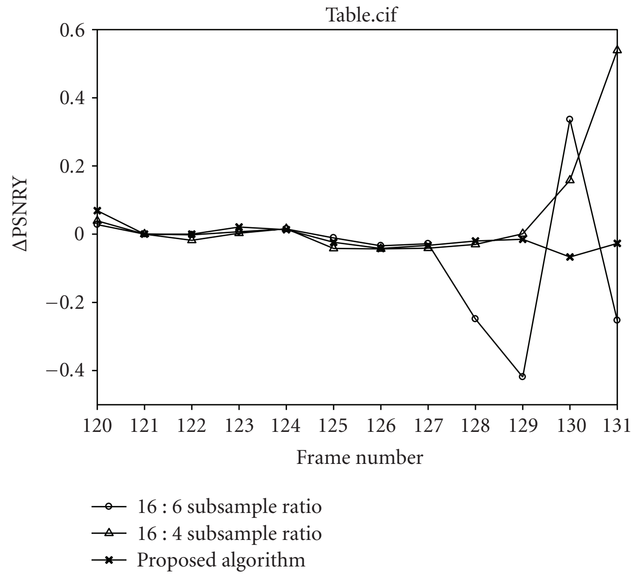


FIGURE 9: The dynamic variation of FSBM quality degradation with fixed subsample ratios and proposed algorithm.

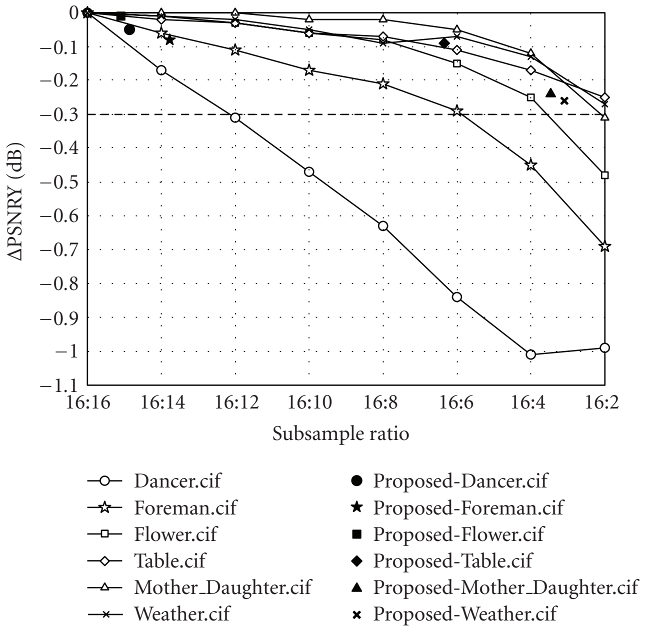


FIGURE 10: The quality degradation chart of FME with fixed subsample ratios and proposed algorithm.

the fixed 16 : 12 subsample ratio to meet the target, but the proposed algorithm can adaptively use lower subsample ratio to save power dissipation while the degradation goal is met. To demonstrate that the proposed algorithm can adaptively select the suitable subsample ratios for each GOP of a tested video sequence, we analyze the average quality degradation of each GOP by using (5) for "table" sequence and the result is shown as in Figure 8. From Figure 8, the first, second, eighth to twentieth GOPs have the lowest degree of high-frequency characteristic and their ZMVCs also show

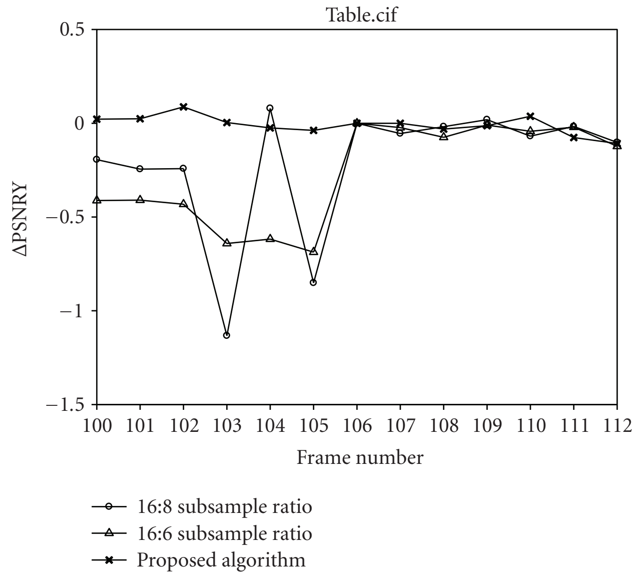


FIGURE 11: The dynamic variation of FME quality degradation with fixed subsample ratios and proposed algorithm.

that they belong to low motion degree, hence these GOPs are allotted 16 : 2 subsample ratio. Moreover, the third GOP has the highest degree of high-frequency characteristic and this GOP is allotted 16 : 16 subsample ratio. The fourth to seventh GOPs also are allotted the suitable subsample ration according to their ZMVCs. Since the tenth GOP is the scene-changing segment, all subsampling algorithms will fail to maintain the quality. Per our simulation with other scene-changing clips, the proposed algorithm does not always miss the optimal ratio. However, in average, the proposed can perform better quality results than the others. Figure 9 shows comparison the PSNRY of each frame using proposed algorithm with the PSNRY of each frame using fixed 16 : 16, 16 : 6, and 16 : 4 subsample ratios. From the analysis result of Figure 9, the PSNRY results of the proposed algorithm is very close to the PSNRY results of fixed 16 : 16 and the proposed algorithm can efficiently save power consumption without affecting visual quality. Finally, to demonstrate the power-saving ability of proposed algorithm, we use (8) to calculate the speedup ratio and the results are shown in Table 6. From Table 6, the speedup ratio can achieve between 1.36 and 5.33. The average speedup ratio is 3.28,

$$\text{Speedup ratio} = \frac{\text{Execution time of FSBM}}{\text{Execution time of simulating VSME}}. \tag{8}$$

The foregoing simulations are implemented using FSBM algorithm in JM10.2 software. Next, the fast motion estimation (FME) algorithm in JM10.2 software is chosen to combine with the proposed algorithm and implement simulations mentioned above again. Table 7 shows results of ΔPSNRY between the proposed algorithm and generic algorithm. Figure 10 shows the distribution diagram of ΔPSNRY versus subsample ratio based on Table 7 and shows that all tested sequences can satisfy to maintain the visual quality degradation under constraint of 0.3 dB. For fast motion sequences, "Dancer," "Foreman," and "Flower," the proposed algorithm can adaptively select low subsample ratio based on their high degree of high-frequency characteristic and visual quality degradation is 0.08 dB at most. Other video sequences are distributed among 16 : 4 and 16 : 2 subsample because low degree of high frequency. Figure 11 shows the PSNR value of each frame for "Table" sequence and the PSNRY results of the proposed algorithm is also very close to the PSNRY results of fixed 16 : 16 subsample ratio. Finally, the results of speedup ratio is shown in Table 8. From Table 8, the speedup ratio can efficiently achieve between 1.056 and 5.428 and operation timing of motion estimation can be more less than FSBM because of less search points. The average speedup ratio is 2.64. Therefore, FME algorithm which combines with the proposed algorithm is a better methodology of motion estimation in H.264/AVC under maintaining the stable visual quality and power-saving for all video sequences.

6. Conclusion

In this paper, we present a quality-stationary ME that is aware of content motion. By setting the subsample ratio according to the motion-level, the proposed algorithm can have the quality degradation low all over the video frames and require low computation load. As shown in the experimental results, with the optimal threshold values, the algorithm can make the quality degradation less than 0.36 dB while saving 69.6% ((1 − 1/3.28) × 100%)) power consumption for FSBM. For the application of FBMA, the quality is stationary with the degradation of 0.27 dB and the power consumption is reduced by the factor of 62.2% ((1 − 1/2.64) × 100%)). The estimation of power consumption reduction is referred to the average subsampling ratio in that the power consumption should be proportional to the subsampling amount. The higher the subsampling amount, the more the power consumption. One can also adjust the size of search range or calculation precision for achieving the quality-stationary. However, either approach cannot have high degree of flexibility for hardware implementation.

Acknowledgment

This work was supported in part by the National Science Council, R.O.C., under the grant number NSC 95-2221-E-009-337-MY3.

References

[1] ISO/IEC CD 11172-2(MPEG-1 Video), "Information technology-coding of moving pictures and asociated audio for digitsl storage media at up about 1.5 Mbits/s: Video," 1993.

[2] ISO/IEC CD 13818-2–ITU-T H.262(MPEG-2 Video), "Information technology-generic coding of moving pictures and asociated audio information: Video," 1995.

[3] ISO/IEC 14496-2 (MPEG-4 Video), "Information Technology-Generic Coding of Audio-Visual Objects," Part2:Visual, 1999.

[4] T. Wiegand, G. J. Sullivan, and A. Luthra, "Draft ITU-T Recommendation H.264 and Final Draft International Standard 14496-10 AVC," VT of ISO/IEC JTC1/SC29/WG11 and ITU-T SG16/Q.6, Doc. JVT-G050r1, Geneva, Switzerland, May 2003.

[5] I. Richardson, *H.264 and MPEG-4 Video Compression*, John Wiley & Sons, New York, NY, USA, 2003.

[6] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, 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.

[7] P. Kuhn, *Algorithm, Complexity Analysis and VLSI Architecture for MPEG-4 Motion Estimation*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.

[8] V. L. Do and K. Y. Yun, "A low-power VLSI architecture for full-search block-matching motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 4, pp. 393–398, 1998.

[9] J.-F. Shen, T.-C. Wang, and L.-G. Chen, "A novel low-power full-search block-matching motion-estimation design for H.263+," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 7, pp. 890–897, 2001.

[10] M. Brünig and W. Niehsen, "Fast full-search block matching," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 2, pp. 241–247, 2001.

[11] L. Sousa and N. Roma, "Low-power array architectures for motion estimation," in *Proceedings of the 3rd IEEE Workshop on Multimedia Signal Processing*, pp. 679–684, 1999.

[12] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Transactions on Communications*, vol. 29, no. 12, pp. 1799–1808, 1981.

[13] E. Ogura, Y. Ikenaga, Y. Iida, Y. Hosoya, M. Takashima, and K. Yamashita, "Cost effective motion estimation processor LSI using a simple and efficient algorithm," *IEEE Transactions on Consumer Electronics*, vol. 41, no. 3, pp. 690–698, 1995.

[14] S. Mietens, P. H. N. De With, and C. Hentschel, "Computational-complexity scalable motion estimation for mobile MPEG encoding," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 281–291, 2004.

[15] M. Chen, L. Chen, and T. Chiueh, "One-dimensional full search motion estimation algorithm for video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 5, pp. 504–509, 1994.

[16] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 438–442, 1994.

[17] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 4, pp. 369–377, 1998.

[18] C. Zhu, X. Lin, and L.-P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 5, pp. 349–355, 2002.

[19] K. B. Kim, Y. G. Jeon, and M.-C. Hong, "Variable step search fast motion estimation for H.264/AVC video coder," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 3, pp. 1281–1286, 2008.

[20] M. G. Sarwer and Q. M. J. Wu, "Adaptive variable block-size early motion estimation termination algorithm for H.264/AVC video coding standard," *IEEE Transactions on*

Circuits and Systems for Video Technology, vol. 19, no. 8, pp. 1196–1201, 2009.

[21] 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.

[22] C. Cai, H. Zeng, and S. K. Mitra, "Fast motion estimation for H.264," *Signal Processing: Image Communication*, vol. 24, no. 8, pp. 630–636, 2009.

[23] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Transactions on Image Processing*, vol. 4, no. 1, pp. 105–107, 1995.

[24] J.-H. Luo, C.-N. Wang, and T. Chiang, "A novel all-binary motion estimation (ABME) with optimized hardware architectures," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 8, pp. 700–712, 2002.

[25] N.-J. Kim, S. Ertürk, and H.-J. Lee, "Two-bit transform based block motion estimation using second derivatives," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 902–910, 2009.

[26] S. Lee and S.-I. Chae, "Motion estimation algorithm using low resolution quantisation," *Electronics Letters*, vol. 32, no. 7, pp. 647–648, 1996.

[27] H. W. Cheng and L. R. Dung, "EFBLA: a two-phase matching algorithm for fast motion estimation," in *Proceedings of the 3rd IEEE Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing*, vol. 2532, pp. 112–119, December 2002.

[28] C.-L. Su and C.-W. Jen, "Motion estimation using MSD-first processing," *IEE Proceedings: Circuits, Devices and Systems*, vol. 150, no. 2, pp. 124–133, 2003.

[29] S.-Y. Huang, C.-Y. Cho, and J.-S. Wang, "Adaptive fast block-matching algorithm by switching search patterns for sequences with wide-range motion content," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 11, pp. 1373–1384, 2005.

[30] K.-H. Ng, L.-M. Po, K.-M. Wong, C.-W. Ting, and K.-W. Cheung, "A search patterns switching algorithm for block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 5, pp. 753–759, 2009.

[31] Y. Nie and K.-K. Ma, "Adaptive irregular pattern search with matching prejudgment for fast block-matching motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 6, pp. 789–794, 2005.

[32] J.-H. Lim and H.-W. Choi, "Adaptive motion estimation algorithm using spatial and temporal correlation," in *Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM '01)*, vol. 2, pp. 473–476, Victoria, Canada, August 2001.

[33] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, no. 2, pp. 148–157, 1993.

[34] C. Cheung and L. Po, "A hierarchical block motion estimation algorithm using partial distortion measure," in *Proceedings of the International Conference on Image Processing (ICIP '97)*, vol. 3, pp. 606–609, October 1997.

[35] C.-K. Cheung and L.-M. Po, "Normalized partial distortion search algorithm for block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 3, pp. 417–422, 2000.

[36] C.-N. Wang, S.-W. Yang, C.-M. Liu, and T. Chiang, "A hierarchical decimation lattice based on N-queen with an application for motion estimation," *IEEE Signal Processing Letters*, vol. 10, no. 8, pp. 228–231, 2003.

[37] C.-N. Wang, S.-W. Yang, C.-M. Liu, and T. Chiang, "A hierarchical N-queen decimation lattice and hardware architecture for motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 4, pp. 429–440, 2004.

[38] H.-W. Cheng and L.-R. Dung, "A vario-power ME architecture using content-based subsample algorithm," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 349–354, 2004.

[39] Y.-L. Chan and W.-C. Siu, "New adaptive pixel decimation for block motion vector estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 1, pp. 113–118, 1996.

[40] Y. K. Wang, Y. Q. Wang, and H. Kuroda, "A globally adaptive pixel-decimation algorithm for block-motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 6, pp. 1006–1011, 2000.

[41] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, Prentice-Hall, Upper Saddle River, NJ, USA, 1999.

[42] Y.-L. Chan and W.-C. Siu, "New adaptive pixel decimation for block motion vector estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 1, pp. 113–118, 1996.

[43] Y. Wang, Y. Wang, and H. Kuroda, "A globally adaptive pixel-decimation algorithm for block-motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 6, pp. 1006–1011, 2000.

[44] L.-R. Dung and M.-C. Lin, "Wide-range motion estimation architecture with dual search windows for high resolution video coding," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E91-A, no. 12, pp. 3638–3650, 2008.

[45] Joint Video Team, "Reference Software JM10.2," http://iphome.hhi.de/suehring/tml/download/old_jm/.

[46] http://www.m4if.org/resources.php#section26.