움직임 방향 연관 및 예측치 적용 기반 적응적 고속 H.264 움직임 추정 알고리즘의 설계

論 文
10-2-2

An Adaptive Fast Motion Estimation Based on Directional Correlation and Predictive Values in H.264

김 정 길*
Cheong-Ghil Kim

Abstract

This research presents an adaptive fast motion estimation (ME) computation on the stage of uneven multi-hexagon grid search (UMHGS) algorithm included in an unsymmetrical-cross multi-hexagon-grid search (UMHexagonS) in H.264 standard. The proposed adaptive method is based on statistical analysis and previously obtained motion vectors to reduce the computational complexity of ME. For this purpose, the algorithm is decomposed into three processes: skipping, terminating, and reducing search areas. Skipping and terminating are determined by the statistical analysis of the collected minimum SAD (sum of absolute difference) and the search area is constrained by the slope of previously obtained motion vectors. Simulation results show that 13%-23% of ME time can be reduced compared with UMHexagonS, while still maintaining a reasonable PSNR (peak signal-to-noise ratio) and average bitrates.

Keywords : Variable block size motion estimation, pipelined array architecture, H.264/AVC, and video coding.

I. Introduction

So far, many approaches for digital video compression have been developed to store or transmit analog data to digital media including H.261, MPEG-1, MPEG-2, H.263, MPEG-4, and H.264. Especially, as information technology advances, both providers and consumers are demanding more complex, quality services, e.g., video telephony on cellular phones and multipoint video conferencing. To satisfy these needs, H.264/AVC standard was approved by the ITU-T as recommendation H.264 and ISO/IEC as International Standard 14496-10 (MPEG-4 Part 10) Advanced Video Coding (AVC) [1].

H.264/AVC is designed for transmitting rectangular video frames with efficient and robust coding, and it can be used for broadcasting and streaming service over packet networks [3]. To achieve low bitrates while maintaining a high video quality, it has adopted several ME methods for motion compensation (MC); variable block-size MC with small block sizes, quarter-sample-accurate MC, and multiple reference picture MC [4]. In general, most standards for digital video compression are based on a hybrid MC/DCT (discrete cosine transform) coding scheme because it can optimally remove the redundancy between frames or in a frame on the temporal and spatial domain. But, in this scheme, the ME and MC tend to consume most of the encoding time [5]. H.264/AVC is also based on a hybrid MC/DCT coding scheme with integer trans-

접수일자 : 2011년 03월 21일
심사일자 : 2011년 05월 17일
수락일자 : 2011년 06월 24일
*교신저자, E-mail : cgkim@nsu.ac.kr

formation instead of DCT. In addition, it contains several methods for good quality of video as stated above. So the ME consumes up to 60% for one reference frame and 80% for 5 reference frames of total encoding time of H.264/AVC codec [6].

As a result, the reduction of the processing time of ME is a prerequisite for real-time applications, such as videophone on mobile devices. So, various fast motion estimation (FME) algorithms have been proposed to reduce the encoding time, especially ME time [6-10]. UMHexagonS [6], Simplified UMHexagonS [7], EZPS (enhanced predictive zonal search) [8], and modified DSR (dynamic search range) [9] are the FME algorithms implemented in JM reference software, namely JM [13]. These algorithms use several methods to reduce the complexity of ME; for instance, they search point predictions, early termination by predicted thresholds, and the hybrid search patterns for UMHexagonS, Simplified UMHexagonS, and EZPS, and they reduce the search range with a motion vector of neighboring blocks for modified DSR. According to the simulation results of [11], modified DSR is useful in some cases without any modification of the original ME structure even though it cannot save much time. Furthermore, with DSR, UMHexagonS and EZPS show the same ME execution time and RD (rate distortion) performance for integer-pel. But, the RD performance of simplified UMHexagonS with DSR is worse even though it is faster than others. Also, EZPS has more switches than UMHexagonS for one block [11]. So, we focused on UMHexagonS and have found that there is still a plenty of room for improving on UMHGS in the search patterns.

In this paper, we propose an adaptive FME algorithm that is improved by statistical analysis and directional correlation in each macroblock. To achieve a good ME performance without degradation of quality, we collect the minimum SAD values in each macroblock and sub-macroblock, and analyze them to skip and terminate several SAD points. From this analysis, we get the four thresholds; the two for skipping and the other two for termination, and those are taken into the account of variable block size in H.264. We also utilize the slope of motion vectors [2] which is already obtained in the previous frames, and the search area is constrained by the direction which is decided by the slope. The proposed algorithm is divided into three sections: skipping the search pattern, reducing search range, and terminating the search pattern. The simulation results show that 13%-23% of ME time can be reduced compared with UMHexagonS while still maintaining reasonable PSNR and average bitrates.

The organization of this paper is as follows. In Section 2, we introduce the UMHexagonS algorithm. Section 3 describes our proposed algorithm. Section 4 presents our simulation results on JM. Conclusions are presented in Section 5.

II. UNSYMMETRICAL-CROSS MULTI-HEXAGON-GRID SEARCH

According to JM, the UMHexagonS algorithm can be classified into four parts as shown in Fig. 1 [13]. In Part I, a search operation is performed at the predicted starting search points. In Part II, the unsymmetrical-cross search pattern is used to cover the general motions. Part III is UMHGS, which is used to cover large and irregular motions. In Part IV, the extended hexagon-based search (EHS) and the small local search (SLS) are employed to refine the motion vector accurately. And the early termination method is used to detect zero blocks [10] during Part III [12,13]. Through the above processes, a search point with the minimum SAD that is obtained in the previous part is used as the starting search point of the following part. In this section, we will introduce Parts I and III that are modified in our proposed algorithm.

1. Prediction of Starting Search Point

In Part I, UMHexagonS algorithm uses four prediction methods to start the ME as the minimum block distortion (MBD) point shown in Fig. 2.

Median prediction determines a starting search

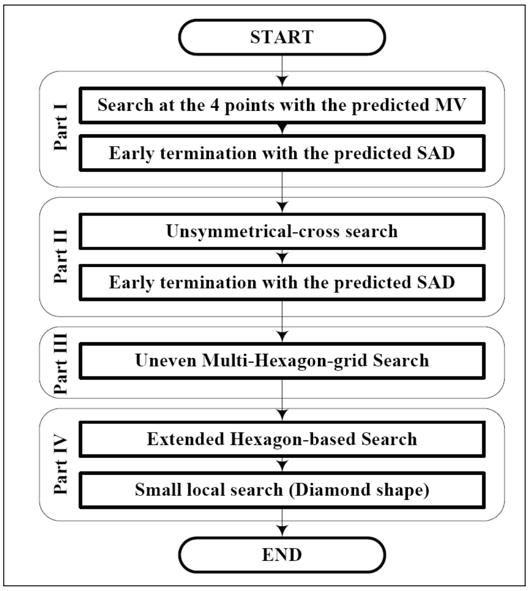

그림 1. UMHexagonS 알고리즘의 실행 단계
Fig. 1. Processing flow of the UMHexagonS algorithm.

point by using the spatial correlation between the current macroblock and its surrounding macroblocks. A median predicted motion vector is the median value in each x and y coordinate of the macroblocks, located on the left, up, and up-right for a current macroblock. Fig. 2(a) shows the median prediction method. If MVleft and/or MVup-right are not available, e.g. lying outside the picture, (0, 0) and/or MVup-left are used to obtain the median predicted vector, respectively.

$$MV_{pred}=median(MV_{left},MV_{up},MV_{up\text{-}right}) \quad (1)$$

Up-layer prediction uses the relationship among 7 modes classified by the block size, i.e., 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, and 4x4. It uses the motion vector of the larger block to get a relative search starting point for its associated smaller blocks. As depicted in Fig. 2(b), Mode 1 is used to predict Modes 2 and 3; Mode 2 is used for Mode 4; Mode 4 is used to predict Modes 5 and 6; and so on.

The corresponding-block prediction selects the motion vector of the macroblock which is placed on the same coordinate of the reference frame just before the current frame as the starting search point. This uses the temporal correlation between the current frame and reference frame. Equation (2) and Fig. 2(c) describe the corresponding block prediction method.

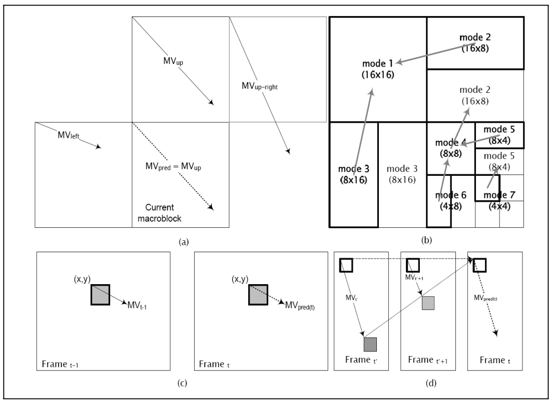

그림 2. 4종류의 예측 모드 : (a) median prediction, (b) up-layer prediction, (c) corresponding-block prediction, and (d) neighboring ref-frame prediction.
Fig. 2. Four types of prediction methods: (a) median prediction, (b) up-layer prediction, (c) corresponding-block prediction, and (d) neighboring ref-frame prediction.

$$MV_{pred(t)} = MV_{t-1} \quad (2)$$

where the *MVpred(t)* is the predicted motion vector for the current macroblock in the current frame *t* and *MVt-1* is the previously obtained motion vector for the macroblock in the reference frame *t-1* just before the current frame.

The neighboring ref-frame prediction uses a temporal correlation between any continuous reference frames, but it can be used only when the reference frame is not the one just before the current frame, i.e., the number of already encoded frames to be referenced has to be more than two. A predicted motion vector for the current macroblock in reference frame *t'* can be obtained by applying the scaling factor to the motion vector of the macroblock in reference frame *t'+1*. The scaling factor is shown in Eq. (3) and Fig. 2(d) shows the details:

$$MV_{pred(t)} = MV_{t'-1} \times \frac{t-t'}{t-t'-1} \quad (3)$$

where the *MVpred(t)* is the predicted motion vector for the current macroblock in the current frame *t*, *MVt'+1* is the motion vector of the macroblock in the reference frame *t'+1*, and *t* and *t'* are the frame numbers of the current frame *t* and the reference frame *t'*, respectively.

2. Uneven Multi-Hexagon-Grid Search

As stated in first paragraph of Section 2, UMHGS is the third part of the UMHexagonS algorithm. Fig. 3 shows the search pattern of UMHGS. UMHGS is divided into two sub-steps. In the first step, as shown in Fig. 3(a), the full search operation is performed with search range, R = 2 (FS), i.e., 25 points. In the second step, the search operation is performed at each of 16 points with an uneven hexagon pattern (UHP) as shown in Fig. 3(b). In this step, the size of UHP expands with R/4 ratio by 1, from 1 to , and this is depicted by the directional arrows from inside to outside in Fig. 3-(b). Therefore, it will be tested by the number of points that are (25 + 16 × R/4), when the number of searching points in the full search with search range, R = 2 and UHP are 25 and 16, respectively. A point with the minimum SAD value will be used as the starting search point of the next part, Part 4.

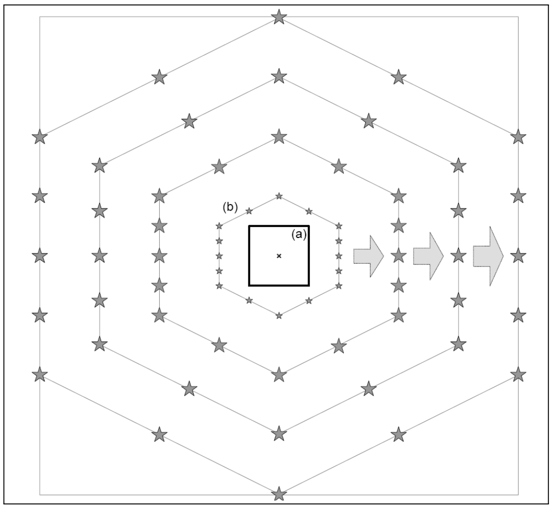

그림 3. Uneven Multi-Hexagon-Grid Search (탐색범위 R = 16): (a) full search with search 탐색범위 2, (b) 16 지점 uneven hexagon pattern

Fig. 3. Uneven Multi-Hexagon-Grid Search (search range R = 16): (a) full search with search range 2, (b) uneven hexagon pattern with 16 points.

III. PROPOSED ALGORITHM

In this paper, we propose an adaptive fast algorithm of integer-pel ME that can reduce the number of SAD operations, while preserving reasonable PSNR and average bitrates. The main idea of the proposed algorithm is to utilize the spatial correlation between the adjacent blocks and the temporal correlation among video frames, and thus we use two factors to design our algorithm; one is predicted motion vectors, called figured motion vectors (MVAF), and their slope of the up-layer and corresponding-block, and the other is a statistically analyzed minimum SAD value that can be used as the threshold.

The proposed algorithm is divided into three parts: skipping based on the statistical threshold (Part A), reducing the search areas with slope of MVAF (Part B), and terminating the iteration earlier with the statistical threshold for expanding UHP (Part C).

The overall process of the proposed algorithm is shown in Fig. 4. In Part A, if the minimum SAD obtained in Part II of UMHexagonS corresponds with the skipping condition, then we can skip the search for an uneven-hexagon pattern. If it does not correspond with the condition, in Part B, the search area of the uneven hexagon pattern is reduced by the direction of motion vectors that was already obtained from the previous encoded frame. Finally, in Part C, we terminate the expansion of the uneven hexagon pattern with the terminating condition. The following sub-sections will describe these processes in detail.

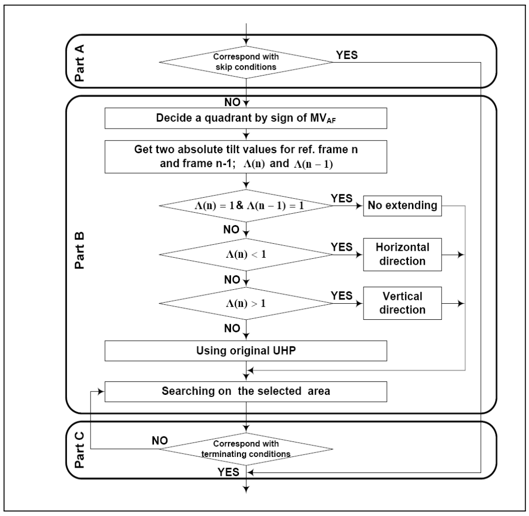

그림 4. 제안 알고리즘의 수행 과정

Fig. 4. The overall process for the proposed algorithm

1. Skipping based on Statistical Analysis

After processing the FS and early termination in UMHGS, non-terminated cases will be processed by an expanding UHP search. By our simulation results, non-terminated cases are 97.53% against the case entering into the FS. In this case, UMHexagonS employs the SAD operation at (16 × R/4) points to cover the large and irregular motions. But, if MVAF is a zero-vector, i.e., (0, 0), the current motion is most likely small. Therefore, if the minimum SAD value for the current block, which is obtained in the first step of UMHGS, is sufficiently small, the possibility that the smaller one appears is low and the current search point can be the MBD point. As a result, the expanding UHP search can be skipped in this case.

To decide whether the expanding UHP search can be skipped or not, we define the threshold, μL, for macroblocks (16x16, 16x8, 8x16) and μS for sub-macroblocks (8x8, 8x4, 4x8) by statistical analysis. We collect the minimum SAD values that are obtained in the result of ME when each MVAF is zero-vector for the macroblock and sub-macroblock. CORRM is defined as the zero corresponding-block predicted MV for the 16x16 size macroblock, UPM as the zero up-layer predicted MV for 16x8 and 8x16 size macroblocks, and UPS as the zero up-layer predicted MV for 8x8, 8x4, and 4x8 size sub-macroblocks. In this part, the 4x4 size sub-macroblock is not considered, because the UMHexagonS algorithm skips Part II and Part III of Fig. 1 when the size of the current block is 4x4. Then we classify the collected data into several intervals by the standard deviation. Fig. 5 shows the distribution of the minimum SAD values in the least significant five intervals; (a), (b), and (c) are the graphs for CORRM, UPM, and UPS, respectively. Each horizontal bar of the graphs in Fig. 5 is composed of five threshold candidates, and we know that the least significant five intervals include 98.88% of the selected minimum SAD values from the accumulated portion of each threshold candidate. So we define the threshold candidate set which is made by combining the least significant five intervals for each case of CORRM, UPM, and UPS and each threshold set is composed of two threshold values, μL and μS for the macroblock and sub-macroblock, respectively. We also simulate the algorithm with each threshold candidate set to monitor the PSNR and average bitrates. The results are listed in Table I which contains a combination of CORRM and UPS. The simulation results show that a larger threshold causes more skipping. Therefore, the thresholds are determined such that μL is 3346 and μS is 357 with negligible degradation of PSNR and average bitrates. We also checked the RD performance to test the suitability of the thresholds, μL and μS, in variable bitrates, and the results are presented in Section IV.

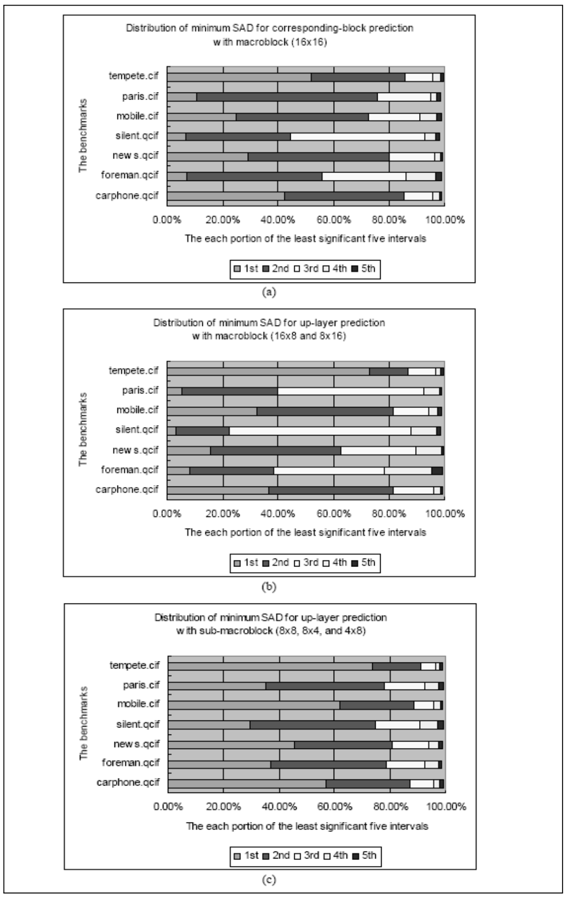

그림 5. 최소 SAD의 분포

Fig. 5. Distributions of the minimum SAD for the least significant five intervals

2. Reducing Search Areas

If the minimum SAD which is obtained in the FS of UMHGS is not small enough to skip, or MVAF is not (0, 0), we cannot ascertain whether it is near the MBD point or not. So we must proceed the expanded UHP search. However, several calculations are required to determine the MBD point. In this section, we will describe a method that reduces the search areas in the expanded UHP. Reducing the search area also uses MVAF that is either from the same location on the previous reference frame or is on the up-layer mode to reduce the search area. Generally, there are strong correlations between macroblocks and/or sub-macroblocks, i.e., one is a temporal correlation when a macroblock is located at the same position for both current and reference frames, the other is a spatial one when a macroblock is partitioned into small size blocks. Therefore, we can reduce the search area by using these characteristics.

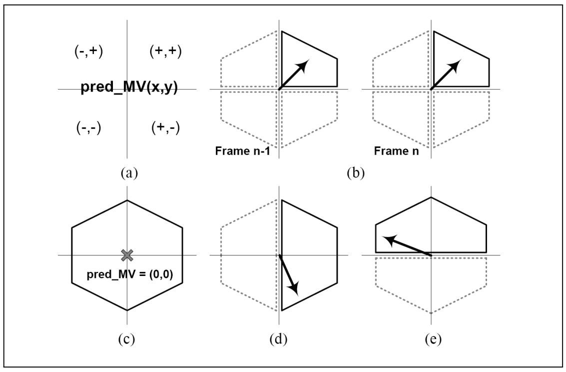

그림 6. 제안 알고리즘의 수행 과정
Fig. 6. Determination of a quadrant and extension of the search areas

Part B of Fig. 4 shows the process of reducing the search area. Part B uses MV_{AF}, and the slope of MV_{AF} to determine and reduce the search area. Fig. 6 shows the details, demonstrating how the search area can be reduced. At first, as shown in Fig. 6(a), a quadrant can be determined by the sign of the x and y components of MV_{AF}, for the current reference frame. After determining a quadrant, the two absolute slope values can be calculated; one is $\Lambda_{(n)}$ for the current reference frame and the other is $\Lambda_{(n-1)}$ for just before the current reference frame in the reference frame list.

$$\Lambda_{(n)} = \frac{|MV_{AFy(n)}|}{|MV_{AFx(n)}|} \quad and \quad \Lambda_{(n-1)} = \frac{|MV_{AFy(n-1)}|}{|MV_{AFx(n-1)}|} \qquad (4)$$

where $MV_{AFx(n)}$ and $MV_{AFy(n)}$ are the x and y components of MV_{AF} in the nth reference frame, $MV_{AFx(n-1)}$ and $MV_{AFy(n-1)}$ are the x and y components of MV_{AF} in the (n-1)th reference frame, and $\Lambda_{(n)}$ and $\Lambda_{(n-1)}$ are the slope of $MV_{AFx(n)}$ and $MV_{AFx(n-1)}$, respectively.

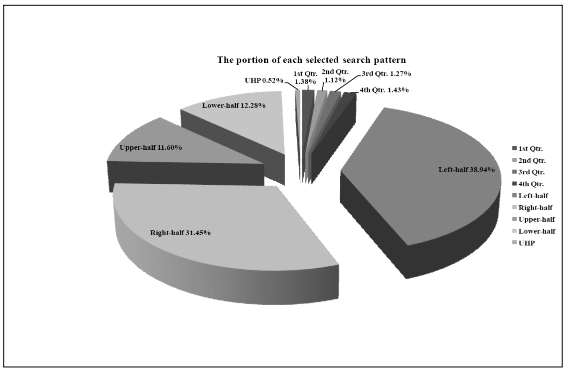

그림 7. 검색 패턴의 구성 비율
Fig. 7. The portions of each selected search pattern

Then, $\Lambda_{(n)}$ and $\Lambda_{(n-1)}$ are compared to determine the search areas. If and is the same as one, the search areas are determined as the quadrants that are selected by the signs of $MV_{AFx(n)}$ and $MV_{AFy(n)}$ as shown in Fig. 6(b). If the slope of the motion vector on the sequence of the reference frames is 1, the slope of the motion vector on the current frame is most likely to be near 1 by correlation of the frames. If those are not 1, we must check the extension of the search areas in the horizontal or vertical direction to get an accurate motion vector. In this case, we also use $\Lambda_{(n)}$ and $\Lambda_{(n-1)}$ to extend the search areas as shown in Fig. 6(d)(e). But, if $MVA_{F(n)}$ is (0, 0), it is hard to extract the correlation between frames or among the macroblocks and sub-macroblocks. Thus, in this case, we select the original UHP (16 points) as shown in Fig. 6(c). Fig. 7 shows the portion of each selected search area in Fig. 6, and it shows that 70% of the motion vectors that are processed in this part tend to move horizontally. As a result, we can perform the search operation to find the MBD point for integer pels; the minimum number of search points is 5 and the maximum number of search points is 16 in each expansion of UHP.

3. Terminating the Enlarging based on Statistical Analyzing

In Section 2, we described how to reduce the search areas. This method will also reduce a large number of SAD operations. But if, in Step B, $MV_{AFx(n)}$ is (0, 0), we use the original UHP (16 points) and the search patterns, which are a quarter of uneven-hexagon, a half of uneven-hexagon, and the original uneven-hexagon, and can be extended in the same manner as in Fig. 3(b). Therefore, we apply the statistical analysis method that is used in Part A to terminate the extension of search patterns that are selected on Part B as depicted in Fig. 6. As seen in Part A in Section 3, if the latest minimum SAD which is obtained in each extended search pattern in Part B of Fig. 4 is small enough, the current search point can also be near the MBD point. Therefore, the extension operation of Fig. 2(b) can be terminated and the next part of UMHexagonS can be performed with the point which has the minimum SAD value. For termination, we also defined the threshold ρL for macroblocks (16x16, 16x8, 8x16) and ρS for sub-macroblocks (8x8, 8x4, 4x8) by statistical analysis, and ρL and ρS are decided as 9250 and 2074, respectively, based on our simulation results. Also, in this part, the threshold ρS does not consider the 4x4 size sub-macroblock, because the UMHexagonS algorithm skips Part II and Part III of Fig. 1 when the size of the current block is 4x4.

IV. EXPERIMENTAL RESULTS

In this paper, we simulate the proposed method with JM reference software of encoder [13] distributed by JVT. The major parameters for encoding are as follows: the GOP structure is IPPP, quantization parameters are 8, 18, 28, and 38, the reference frame number is 5, the search range equals 32, and the hadamard transform and RD Optimization are used. All the other parameters were set as the Baseline profile. The benchmarks for the experiments are QCIF and CIF, test sequences that are generally used. In particular, because the major purpose of the slope is to determine the direction of the motion vector, we used a difference operation instead of the division operation to avoid the floating-point operations for slope calculation in Part B of the proposed algorithm.

$$Time(UMHGS) = \left(\frac{TIME_{\mathrm{Proposed}}}{TIME_{UMHexagonS}} \times 100 \right) \qquad (5)$$

Table 1 shows the proportion of variable measured times and the difference of PSNR and average bitrates in each benchmark for comparison between UMHexagonS and the proposed algorithm when the quantization parameters are changed among 8, 18, 28, and 38. In this table, “TOTAL” means the proportion of total encoding time, “ME” means the proportion of ME time, and “UMHGS” means the proportion of ME time of the proposed algorithm in UMHGS against one of the UMHexagonS algorithms, and it is obtained by Eq. (5) where TIMEProposed and TIMEUMHexagonS are motion estimation time in UMHGS. “PSNR” and “Bitrates” represent the loss or the gain of PSNR and average bitrates, respectively, and they are calculated by the difference between the proposed algorithm and UMHexagonS algorithm. If the “PSNR” is a positive value, the quality of encoded video frames with the proposed algorithm is improved. On the other hand, if “Bitrates” is a negative value, the proposed algorithm needs fewer bits than the UMHexagonS algorithm to encode a video frames.

In Table 1, we can see that the combination of Part A, Part B, and Part C in the proposed algorithm reduce 98% of ME time in UMHGS, 19% of ME time in overall ME process, and 9% of encoding time in the total encoding process with a negligible degradation of PSNR and average bitrates. The proposed algorithm works effectively when the encoded video frames have a large and complex motion, e.g., ‘foreman’, ‘football’, ‘mobile’, and ‘stefan’.

표 1. 실험 결과

Table 1. Simulation Results (5 Reference Frames, Search range = 32, and Frequency = 30HZ).

Benchmarks	Type	Frames	QP	TOTAL	ME	UMHGS	PSNR (dB)			Average Bitrates (kbits/sec)
							Y	U	V	
carphone	QCIF	381	8	87.28%	74.59%	1.02%	0.00	-0.01	0.01	0.67
			18	87.21%	76.48%	1.42%	0.00	0.00	0.05	0.36
			28	88.95%	80.58%	1.28%	0.00	-0.03	0.01	0.36
			38	91.02%	83.39%	1.81%	-0.01	0.02	-0.13	-0.11
foreman		399	8	85.51%	72.59%	0.83%	0.00	0.00	0.01	1.08
			18	85.42%	73.14%	0.72%	0.00	-0.01	0.01	0.13
			28	86.97%	77.52%	0.98%	0.01	0.01	-0.01	0.08
			38	89.31%	80.73%	0.92%	-0.02	0.08	0.03	0.07
grandma		299	8	94.58%	84.56%	2.11%	0.00	-0.01	-0.01	-0.07
			18	94.90%	88.02%	1.86%	-0.01	0.00	0.00	-0.21
			28	96.11%	92.09%	0.34%	0.00	0.04	0.01	-0.08
			38	97.15%	95.91%	0.00%	0.00	-0.02	-0.04	0.10
highway		299	8	89.25%	74.42%	0.89%	0.00	0.00	0.00	1.27
			18	90.61%	81.90%	1.34%	0.00	0.00	-0.01	1.15
			28	94.51%	90.23%	1.22%	-0.03	0.02	0.01	0.17
			38	96.39%	93.88%	0.00%	0.03	-0.01	-0.02	0.15
mthr_dotr		299	8	90.67%	78.99%	0.67%	-0.01	0.00	0.00	-1.02
			18	91.10%	83.04%	1.64%	-0.01	0.00	0.01	-0.39
			28	93.96%	86.69%	1.04%	0.01	-0.02	0.05	0.18
			38	96.23%	93.01%	1.26%	0.00	-0.07	-0.04	0.15
news		299	8	94.66%	84.51%	1.27%	0.00	0.00	0.01	-0.01
			18	93.97%	89.02%	0.85%	0.00	0.00	-0.04	-0.30
			28	94.47%	89.79%	1.20%	0.02	0.02	-0.04	-0.14
			38	94.70%	89.26%	1.32%	0.05	0.05	0.06	0.14
salesman		299	8	95.12%	86.06%	0.42%	-0.01	-0.01	0.01	1.35
			18	94.99%	86.26%	0.88%	-0.01	-0.03	-0.01	-0.22
			28	95.59%	88.51%	0.53%	0.00	0.04	0.01	0.18
			38	96.05%	91.75%	0.97%	-0.01	0.09	0.00	0.04
silent		299	8	91.71%	79.16%	1.43%	0.01	0.01	0.02	1.49
			18	91.52%	82.13%	0.76%	-0.01	0.02	0.01	0.38
			28	92.28%	82.61%	1.27%	0.03	-0.03	-0.06	0.14
			38	93.77%	90.49%	0.64%	-0.03	-0.02	0.02	0.01
football	CIF	89	8	80.36%	66.39%	1.44%	0.00	0.00	0.00	13.82
			18	79.12%	68.42%	1.11%	0.01	0.00	0.00	10.23
			28	80.77%	71.11%	1.76%	-0.01	-0.03	0.01	6.93
			38	83.46%	73.22%	2.86%	0.00	0.02	0.10	4.76
highway		299	8	90.50%	78.38%	0.53%	0.00	0.00	0.00	-0.31
			18	89.41%	79.82%	0.76%	0.00	0.00	0.00	0.82
			28	94.24%	87.46%	0.94%	-0.01	-0.01	0.01	1.26
			38	96.78%	92.61%	2.17%	-0.02	0.00	-0.03	0.41
mobile		299	8	85.11%	68.25%	1.02%	0.00	0.00	-0.01	-2.24
			18	84.34%	67.46%	1.01%	0.00	0.00	-0.01	-0.17
			28	84.76%	71.80%	0.98%	0.00	0.00	0.00	-1.34
			38	85.95%	75.96%	1.71%	0.02	0.02	0.01	0.20
paris		299	8	94.01%	82.84%	0.89%	0.00	0.00	-0.01	0.30
			18	93.46%	84.06%	1.11%	0.00	0.00	-0.01	-0.27
			28	94.22%	85.96%	1.24%	0.01	0.01	0.02	0.48
			38	94.58%	88.41%	1.44%	0.00	0.00	-0.01	0.24
stefan		89	8	86.65%	70.91%	1.15%	0.00	0.00	0.00	1.96
			18	86.33%	73.75%	1.02%	0.00	-0.01	0.01	4.51
			28	86.62%	74.60%	1.77%	-0.01	0.00	0.01	3.31
			38	87.14%	78.51%	1.24%	-0.04	0.04	0.02	0.16
tempete		259	8	84.99%	69.42%	0.92%	0.00	-0.01	0.00	0.29
			18	84.50%	70.15%	1.10%	0.00	0.00	0.00	1.43
			28	85.76%	74.04%	1.02%	0.00	-0.01	0.00	0.83
			38	88.25%	79.42%	1.64%	-0.01	0.01	-0.01	-0.12
Average			8	89.31%	76.50%	1.04%	0.00	0.00	0.00	1.33
			18	89.06%	78.83%	1.11%	0.00	0.00	0.00	1.25
			28	90.66%	82.36%	1.11%	0.00	0.00	0.00	0.88
			38	92.20%	86.18%	1.28%	0.00	0.02	0.00	0.44

V. CONCLUSION

In this paper, we propose an adaptive fast ME algorithm that reduces the block matching points by using a statistically analyzed minimum SAD value and a previously obtained motion vector in the H.264 encoder. As a result, the proposed algorithm employs skipping and terminating thresholds and predicts the possible area where the MBD point can be taken as the reference motion vector. Simulation results show that motion estimation time is reduced by 13%-23% by using skipping, a constrained search area, and a terminating process, without a significant degradation of video quality.

Acknowledgement

The funding for this research was made by Namseoul University in 2011.

[References]

[1] Advanced Video Coding for Generic Audiovisual Services, ISO/IEC 14496-10:2005(E) ITU-T Rec. H.264(E), 2005.

[2] C. G. Kim, I. J. Lee, and S. D. Kim, "Reduced Uneven Multi-Hexagon-Grid Search for Fast Integer Pel Motion Estimation in H.264/AVC," *The International Conference on Image Analysis and Recognition (ICIAR 2007), LNCS, 4633*, pp. 708 -714, 2007.

[3] I. E. G. Richardson, H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia, Wiley, 2004.

[4] 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.

[5] T. Sikora, "Trends and perspectives in image and video coding," *Proceedings of IEEE,* vol. 93, no. 1, pp. 6-17, 2005.

[6] JVT-F017, Fast integer pel and fractional pel motion estimation, 2002.

[7] JVT-P021, Improved and simplified fast motion estimation for JM, 2005.

[8] JVT-E023, Fast Motion Estimation within the JVT codec, 2002.

[9] JVT-Q088, Modification of Dynamic Search Range for JVT, 2005.

[10] 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.

[11] JVT-Q089, Comments on Motion Estimation Algorithms in Current JM Software, 2005.

[12] JVT-G016, Fast Motion Estimation for JVT, 2003.

[13] H.264/AVC Reference Software JM 11.0 2006 [Online]. Available: http://iphome.hhi.de/suehring/tml/

Biography

<그림서식>
1.크기 :
22mm*28mm
2.위치 : 어울림
3.안쪽여백 :
상하좌우 0mm
3.바깥쪽여백: 오른쪽
2mm, 아래쪽 1mm
4.테두리 있음

Cheong Ghil Kim

1987 B.S. in Computer Science, at University of Redaldns, U.S.A.

2003 M.S. in Computer Science, at Yonsei University, Korea

2006 Ph.D. in Computer Science, Yonsei University, Korea

2006~ 2008 PostDoc and Research Professor at the Dept. of Computer Science, Yonsei University

2008 ~ Current Professor at Namseoul University

<Research Areas> Mobile Embedded Systems, Parallel Processing

<e-mail> cgkim@nsu.ac.kr