# H.264/AVC 움직임 추정을 위한 효율적인 정적 블록 스킵 방법과 결합된 다이아몬드 웹 격자 탐색 알고리즘

# A Diamond Web-grid Search Algorithm Combined with Efficient Stationary Block Skip Method for H.264/AVC Motion Estimation

정 창 욱* 최 진 구** 이케나가 다케시*** 고토 사토시****
Chang-Uk Jeong Jin-Ku Choi Takeshi Ikenaga Satoshi Goto

## 요 약

H.264/AVC 표준은 여러 가지 신기술들을 접목시킴으로써 기존의 동영상 표준들보다 한층 개선된 부호화 효율성을 제공한다. 하지만, H.264/AVC 인코더의 향상된 부호화 기술은 그것의 전반적인 복잡도를 크게 증가시켰다. 따라서, 인코더의 복잡도 수준을 경감시키기 위한 최적화의 연구는 중대한 선결 과제이다. 특히, 움직임 추정 부분에 대한 계산량의 비율은 인코더의 작업시간을 크게 좌우한다. 본 논문에서는 완전 다이아몬드와 12각형을 기본 탐색 패턴으로 사용하고 특정한 임계기준치를 적용시킴으로써 효율적으로 정적 블록들을 스킵하는 다이아몬드 웹 격자 탐색 알고리즘을 제안한다. 실험 결과는 본 논문에서 제안된 기법이 기존의 UMHexagonS 알고리즘의 계산량을 12%까지 감소시키면서도 유사한 PSNR을 유지한다는 것을 보여준다.

## ABSTRACT

H.264/AVC offers a better encoding efficiency than conventional video standards by adopting many new encoding techniques. However, the advanced coding techniques also add to the overall complexity for H.264/AVC encoder. Accordingly, it is necessary to perform optimization to alleviate the level of complexity for the video encoder. The amount of computation for motion estimation is of particular importance. In this paper, we propose a diamond web-grid search algorithm combined with efficient stationary block skip method which employs full diamond and dodecagon search patterns, and the variable thresholds are used for performing an effective skip of stationary blocks. The experimental results indicate that the proposed technique reduces the computations of the unsymmetrical-cross multi-hexagon-grid search algorithm by up to 12% while maintaining a similar PSNR performance.

☞ KeyWords : motion estimation, H.264 encoder, block-matching algorithm, diamond search, web-grid search
움직임 추정, H.264 인코더, 블록 정합 알고리즘, 다이아몬드 탐색, 웹 격자 탐색

## 1. Introduction

Despite recent advances in storage devices and transmission networks, rapidly increasing video file sizes are prompting further developments of compression technology and video compression standards such as ISO/IEC MPEG-1, MPEG-2, MPEG-4, ITU-T H.261, H.263 [1], and the most recent H.264 [2], [3]. H.264/AVC is the latest in video compression standards that deploy encoding technologies such as variable block-size motion estimation (VBSME), multiple reference frames, 4×4 integer DCT, and 1/4 pixel motion estimation (ME) and motion compensation (MC). Thanks to many advanced encoding techniques, H.264/AVC encoder

* 정 회 원 : 일본 와세다대학교 GITS연구과 박사과정
jeongchanguk@gmail.com
** 정 회 원 : 한국산업기술대학교 컴퓨터공학과 교수
jkchey@kpu.ac.kr
*** 정 회 원 : 일본 와세다대학교 IPS연구과 교수
ikenaga@waseda.jp
**** 정 회 원 : 일본 와세다대학교 IPS연구과 교수
goto@waseda.jp
[2009/11/12 투고 – 2009/11/16 심사 – 2009/12/16 심사완료]

is able to provide twice the compression rate that MPEG-4 does for an identical video quality. However, adopting new encoding techniques significantly increases the amount of computation. In particular, motion estimation accounts for the greatest portion of H.264/AVC's encoding time. Accordingly, optimizing the motion estimation algorithm is very important for alleviating the computing complexity in H.264/AVC encoder.

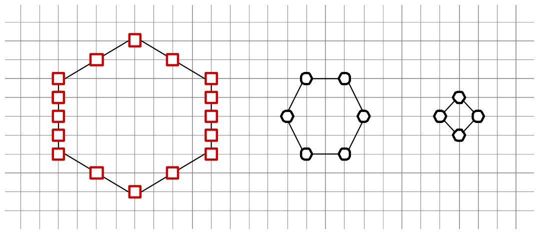
Motion estimation is an effective and representative technique for removing the temporal redundancy in video sequences. Block-based motion estimation has been adopted by most video compression standards including H.264/AVC. Maybe the most popular technique for block-based motion estimation is the block-matching algorithm (BMA). BMA is widely used in video compression standards due to its efficient prediction, accurate estimation, and simple implementation. Working under the assumption that every pixel in a block has an identical motion vector (MV) that displays the translational motion, BMA compares the block of the current frame with that of the reference frame to determine an optimal block. However, restricted search ranges (SR) and limited block sizes for reducing the calculations during real-time implementation undermine the accuracy and reliability of estimation. As the simplest BMA, the full search (FS) algorithm searches exhaustively all reference blocks within the search range. Although the FS provides outstanding quality performance and simple data flow and circuit control, a prohibitive amount of computation is required if the search range becomes too large. This high computational complexity makes it often not suitable for real-time implementations.

Over the past few years, many fast algorithms such as the three-step search (3SS) [4], the new three-step search (N3SS) [5], four-step search (4SS) [6], block-based gradient descent search (BBGDS) [7], diamond search (DS) [8], [9], hexagon-based search (HEXBS) [10], cross-diamond search (CDS) [11], efficient three-step search (E3SS) [12], and cross-diamond-hexagonal search (CDHS) [13] have been proposed to replace FS algorithm with a high level of computational complexity. Each BMA uses a different search strategy to satisfy the accuracy of motion estimation and search speed. The unsymmetrical-cross multi-hexagon-grid search (UMHexagonS) [14] algorithm has been recently proposed as a reference ME for the H.264/AVC JM software [15]. The UMHexagonS is a high-performance hybrid motion estimation algorithm that combines many improved technologies, but is significantly slower than other fast BMAs.

In this paper, we propose a hybrid motion estimation algorithm to further improve the performance of UMHexagonS. The remaining sections of this paper are constructed as follows. Section 2 reviews the UMHexagonS algorithm. In section 3, real-world video sequences are analyzed. In section 4, our proposed methods are presented. The performance can be verified based on experimental results in section 5. Finally, in section 6, the conclusions of this paper will be given.

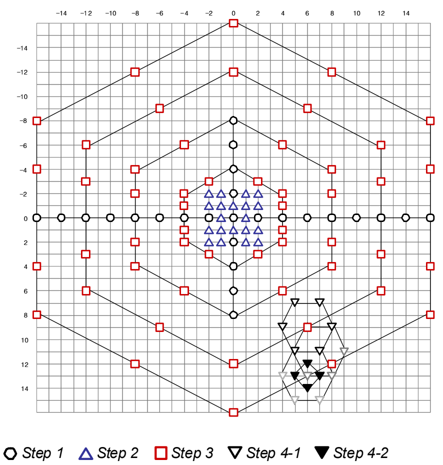
## 2. Unsymmetrical-cross Multi-hexagon-grid Search Algorithm

The UMHexagonS algorithm embedded in the H.264/AVC reference software JM is a high-performance hybrid motion estimation algorithm that combines many improved techniques, including motion vector prediction and early termination, and search patterns of cross, diamond, and hexagon. As shown in Figure 1 (a), a 16 points big hexagon pattern is based on the fact that generally horizontal motion is much heavier than that of vertical motion for real-world video. Two search patterns of Figure 1 (b) and (c) are adopted in the extended hexagon-based search (EHS) [14], which is used as a sub-algorithm in the UMHexagonS. The search process of the UMHexagonS algorithm is summarized as follows:



(a) (b) (c)

(Fig. 1) Search patterns used in UMHexagonS.
(a) 16 points big hexagon pattern,
(b) Hexagon search pattern,
(c) Small diamond search pattern.




(Fig. 2) Search process of UMHexagonS.

START: Before the integer-pixel motion estimation, the start search point should be decided at first.

- *Early Termination.*

STEP 1: First, an unsymmetrical-cross search is made. The spacing between checking points is two and there are twice as many points horizontally than vertically.

- *Early Termination.*

STEP 2: Starting from the best point found in the cross search step, next small rectangular full search is made.

- *Early Termination.*

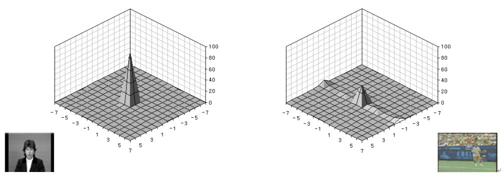
STEP 3: A multi-hexagon-grid search strategy is taken (from 1 to SR/4). The 16 points hexagon pattern used in the grid search has more points at the left and right edges.

- *Termination by Threshold for Strong Motion.*

STEP 4: In the last step, EHS (STEP 4-1) or small diamond search (STEP 4-2) is used to refine the best MV found in the multi-hexagon-grid stage.

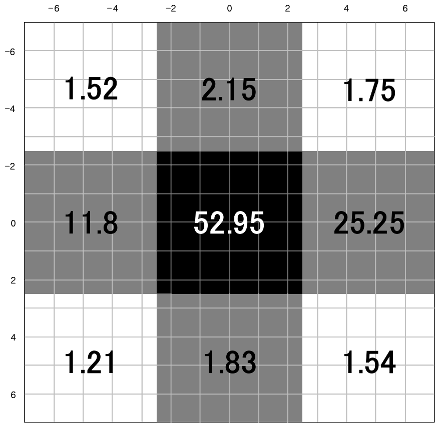
## 3. Motion Vector Distribution

Figure 3 and 4 show the motion vector probability (MVP) distributions resulted from simulations using the full search algorithm with a search range SR = ±7. Sequences used for Figure 3 (a) and (b) are CIF Sequence “Claire” (100 frames) and CCIR601 Sequence “Stefan” (200 frames) respectively. Figure 4 is using three CCIR601 Sequences – “Garden”, “Stefan”, and “Susie” (30 frames respectively).



(a) (b)

(Fig. 3) Motion vector distributions for
(a) “Claire” and (b) “Stefan”.




(Fig. 4) Motion vector probability distribution for three CCIR601s.

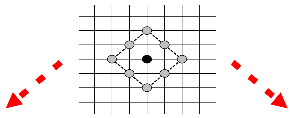
In Figure 3 (a), over 89% of motion vectors are found at the central point (0, 0). Within the central 5×5 area, most of motion vectors are also found. The MVP distribution for the CCIR601 sequence “Stefan” including tough and rough sports scene is shown in Figure 3 (b). Unlike the distribution as illustrated in Figure 3 (a), Figure 3 (b) does not show altogether center-biased MVP distribution but wide distribution.

Figure 4 displays the results of MV’s accumulated distribution for the three CCIR601 sequences. The MVP distribution diagram can be divided into nine regions to comprehensively analyze the general MVP distribution tendency of video data. In Figure 4, it can be seen that the portion of MV within the 5×5 region in the center of the search range is the highest at 53%. Moreover, the portions of MVP in the horizontal and vertical regions are especially high in the regions other than the central region. The major cause for this phenomenon is closely related to the fact that the camera capturing the images mainly moves in horizontal and vertical directions.

# 4. Proposed Methods

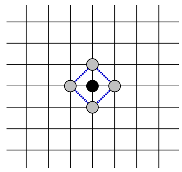
## 4.1 Revised Diamond Search Algorithm

### 4.1.1 Search Patterns and Search Process of RDS



(a) Large diamond-shaped pattern (LDSP) of DS

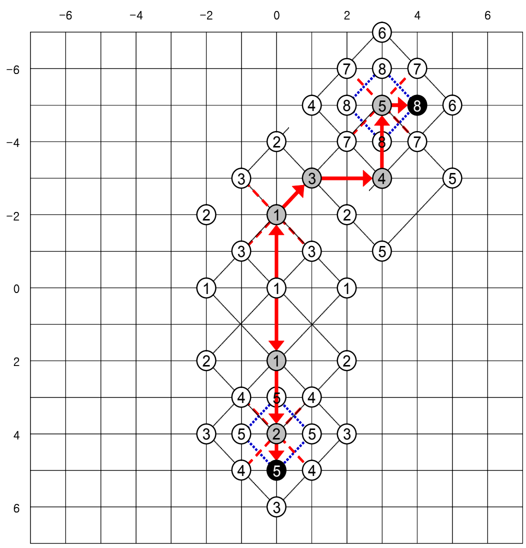
(b) Redefined LDSP (RLDSP)
(c) Small X-shaped pattern (SXSP)



(d) Small diamond-shaped pattern (SDSP)
(Fig. 5) Redefined search patterns from LDSP used in original DS.

The dense search points in the large diamond-shaped pattern (LDSP) of the original DS algorithm, as shown in Figure 5 (a), are inadequate for searching wide-area movements. On the other hand, in Figure 5 (b), the redefined large diamond-shaped pattern (RLDSP) of the distributed search points is more appropriate for horizontal and vertical searches. Furthermore, the small X-shaped pattern (SXSP) as shown in Figure 5 (c), which has branched out from the LDSP, can flexibly cope with local minimum block distortion measure (BDM) points. The small diamond-shaped pattern (SDSP) as shown in Figure 5 (d) is used in the final searching step of the proposed revised diamond search (RDS) algorithm. Two Examples of search paths of the proposed RDS are shown in Figure 6. The search process of the RDS algorithm using RLDSP, SXSP, and SDSP can be summarized as follows:

STEP 1: The 5 checking points of RLDSP are tested. If the minimum BDM point is located at the center, go to STEP 3; otherwise, go to STEP 2.
STEP 2: A new RLDSP with the center at the previous minimum distortion point is formed. If the minimum BDM point obtained is located at the center of RLDSP, go to STEP 3; otherwise, recursively repeat this step.
STEP 3: Switch the search pattern from RLDSP to SXSP. If the minimum BDM point occurs at the center of SXSP, go to STEP 4; otherwise, repeat STEP 2.




(Fig. 6) Two examples of search paths of RDS.

STEP 4: One of the 5 checking points of SDSP is determined as the new minimum BDM point, which is the final MV.

### 4.1.2 Performance Evaluation of RDS

The performance evaluation experiments of the proposed RDS algorithm evaluate the search speed and quality. The search speed can be checked with the average number of search points per block, and the average PSNR (peak signal-to-noise ratio) is used to evaluate the search quality. A total of eight sequences, named “Claire” (CIF 360×288, 100 frames), “Coastguard” (CIF 352×288, 200 frames), “Football” (CIF 360×240, 200 frames), “Salesman” (CIF 360×288, 200 frames), “Football” (CCIR601 720×486, 50 frames), “Garden” (CCIR601 720×486, 90 frames), “Stefan” (CCIR601 720×480, 90 frames), and “Susie” (CCIR601 720×486, 90 frames), are used for the experiment. The sum of absolute differences (SAD) criteria is used for BDM measurement. It is defined as follows:

$$SAD = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} | C_{ij} - R_{ij} | \quad (1)$$

where N is the parameter about the macroblock size. Cij and Rij are current and reference area samples respectively. The block size and the search range are set at 16×16 and SR = ±15, respectively.

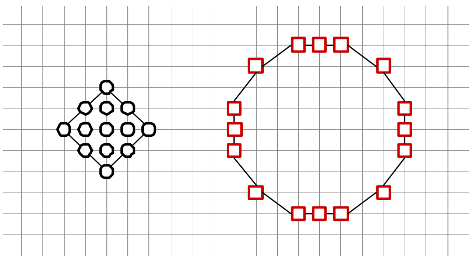
Tables 1 and 2 compare the performance of the proposed RDS algorithm with those of BMAs such as DS, HEXBS, and CDS. The results in Table 2 indicate that the proposed RDS always uses less search points than the original DS. However, the RDS provides a higher prediction quality than the DS in most sequences, as shown in Table 1. In particular, the search quality is more remarkable for CCIR601 sequences. In sequences like “Football” and “Stefan” that include intense and abrupt movements, the proposed algorithm outperforms the DS algorithm in terms of the search speed and quality.

(Table 1) PSNR comparison.

| BMA | CIF Sequences | | | | Average |
|---|---|---|---|---|---|
| | Claire | Coastguard | Football | Salesman | |
| DS | 41.162 | 29.483 | 24.241 | 35.281 | 32.542 |
| HEXBS | -0.325 | -0.161 | -0.205 | -0.043 | -0.184 |
| CDS | -0.015 | -0.005 | -0.051 | 0.096 | 0.006 |
| RDS | -0.023 | -0.027 | 0.032 | -0.010 | -0.007 |
| BMA | CCIR601 Sequences | | | | Average |
| | Football | Garden | Stefan | Susie | |
| DS | 24.643 | 26.252 | 22.616 | 34.659 | 27.043 |
| HEXBS | -0.103 | -0.194 | 0.383 | 0.113 | 0.050 |
| CDS | -0.043 | -0.038 | -0.058 | -0.097 | -0.059 |
| RDS | 0.042 | 0.093 | 0.611 | 0.172 | 0.230 |

(Table 2) Search points comparison.

| BMA | CIF Sequences | | | | Average |
|---|---|---|---|---|---|
| | Claire | Coastguard | Football | Salesman | |
| DS | 13.226 | 17.692 | 23.898 | 13.831 | 17.162 |
| HEXBS | -2.107 | -3.784 | -6.436 | -2.522 | -3.712 |
| CDS | -3.651 | 1.157 | 0.555 | -3.203 | -1.286 |
| RDS | -0.033 | -1.680 | -3.198 | -0.240 | -1.288 |
| BMA | CCIR601 Sequences | | | | Average |
| | Football | Garden | Stefan | Susie | |
| DS | 23.051 | 22.328 | 20.491 | 20.463 | 21.583 |
| HEXBS | -6.467 | -5.767 | -4.548 | -5.037 | -5.455 |
| CDS | 0.240 | 3.221 | -0.113 | 0.585 | 0.983 |
| RDS | -3.053 | -3.355 | -2.004 | -2.140 | -2.638 |



(a) (b)

(Fig. 7) New search patterns used in DWS.
(a) Full diamond search pattern,
(b) Dodecagon search pattern.

## 4.2 Diamond Web-grid Search lgorithm

### 4.2.1 Search Patterns and Search Process of DWS

The proposed diamond web-grid search (DWS) algorithm for H.264/AVC motion estimation was developed based on the MVP distribution and multiple local minima for real-world sequences. The full diamond search pattern and dodecagon search pattern, as shown in Figure 7 (a) and (b), are used as basic search patterns in the proposed DWS algorithm. The full diamond search pattern is very suitable for searching center-biased motions. In contrast to the multi-hexagon-grid used in the UMHexagonS, the multi-dodecagon-grid (web-grid) is considering all directions, but is heavier in horizontal and vertical directions than in diagonal directions. In the last searching step of the DWS, the RDS algorithm proposed in section 4.1 can be used to refine the best MV. The DWS algorithm can be summarized in three steps, as follows:

START: Before the integer-pixel motion estimation, the start search point should be decided at first.

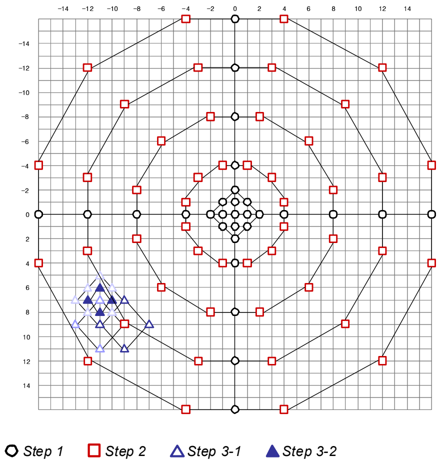
- *Early Termination.*

STEP 1: First, a full diamond search and a symmetrical-cross search are made. The spacing between checking points of the symmetrical-cross search pattern is 4.

- Early Termination.

STEP 2: A web-grid search strategy is taken. The 16 points dodecagon pattern is scaled to various sizes (from 1 to SR/4).

- Termination by Threshold for Strong Motion.




(Fig. 8) Search process of DWS.

STEP 3: In the final step, an unrestricted center-biased search (STEP 3-1: RDS or EHS) or the small diamond search (STEP 3-2) is adopted to refine the best MV found in the web-grid stage. The search process of the proposed DWS algorithm is also shown in Figure 8.

### 4.2.2 Performance Evaluation of DWS

The experiments are conducted by using four QCIF sequences “Claire”, “Container”, “Grandma”, and “Salesman” and four CIF sequences “Coastguard”, “Mobile”, “News”, and “Stefan”. Three hundred frames are used for each of the eight sequences. We compare the proposed algorithms with fast full search (FFS) and UMHexagonS algorithm using the H.264/AVC reference software JM12.4 with search range SR = 16, quantization parameter QP = 40, and rate distortion optimization RDO = 0, and the baseline profile. DW+EHS and DW+RDS algorithms, in STEP 3-1 of the DWS, are using EHS and RDS respectively.

Tables 3 and 4 show that the proposed DW+RDS is about 0.011 dB higher compared with that of the UMHexagonS in terms of PSNR while the average bit rate increase is 0.02%. From Table 5, the motion estimation time reduction is still about 5%.

(Table 3) PSNR [dB] comparison.

| ME | | FFS | UMHS | DW+EHS | DW+RDS |
|---|---|---|---|---|---|
| QCIF | Claire | 31.477 | 31.384 | 0.017 | 0.067 |
| | Container | 27.850 | 27.841 | -0.014 | -0.015 |
| | Grandma | 29.683 | 29.621 | -0.001 | 0.033 |
| | Salesman | 27.381 | 27.367 | -0.010 | 0.013 |
| CIF | Coastguard | 26.675 | 26.663 | -0.010 | -0.016 |
| | Mobile | 24.380 | 24.342 | 0.002 | -0.004 |
| | News | 29.902 | 29.864 | 0.002 | -0.006 |
| | Stefan | 26.372 | 26.339 | -0.003 | 0.015 |
| Average | | - | - | -0.002 | 0.011 |

(Table 4) Bit rate [%] comparison.

| ME | | FFS | UMHS | DW+EHS | DW+RDS |
|---|---|---|---|---|---|
| QCIF | Claire | 7593 | 7528 | -0.11 | 0.05 |
| | Container | 8402 | 8414 | -0.56 | -0.57 |
| | Grandma | 6099 | 6067 | -0.26 | -0.36 |
| | Salesman | 9660 | 9510 | 0.72 | 0.81 |
| CIF | Coastguard | 111963 | 111654 | 0.41 | 0.31 |
| | Mobile | 228985 | 229682 | -0.33 | -0.38 |
| | News | 49051 | 48988 | -0.36 | -0.21 |
| | Stefan | 240217 | 232526 | 0.39 | 0.55 |
| Average | | - | - | -0.01 | 0.02 |

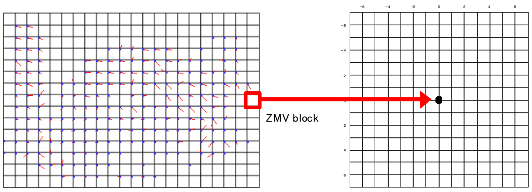
(Table 5) Motion estimation time [%] comparison.

| ME | | FFS | UMHS | DW+EHS | DW+RDS |
|---|---|---|---|---|---|
| QCIF | Claire | 45912 | 9017 | -9.69 | -7.04 |
| | Container | 46505 | 10792 | -3.90 | -4.36 |
| | Grandma | 47674 | 10085 | -5.90 | -6.09 |
| | Salesman | 47576 | 12073 | -6.92 | -3.98 |
| CIF | Coastguard | 196767 | 55820 | -5.08 | -5.33 |
| | Mobile | 189266 | 60204 | -4.63 | -5.69 |
| | News | 186276 | 39662 | -4.98 | -2.98 |
| | Stefan | 189677 | 57613 | -6.21 | -6.96 |
| Average | | - | - | -5.92 | -5.30 |

## 4.3 Efficient Stationary Block Skip ethod

### 4.3.1 Skipping Zero Motion Vector

A block with a zero motion vector (ZMV) is regarded as a stationary block. ZMVs are usually distributed around the background of a video sequence. In Figure 9, a ZMV is positioned at the center of the search window. Most of BMAs start the search process from the central point of the search window. Attention needs to be paid to the fact that ZMV only uses a single central point, and that the BDM value from the previous frame can be A Diamond Web-grid Search Algorithm Combined with Efficient Stationary Block Skip Method for H.264/AVC Motion Estimation reused. SAD or SATD (sum of absolute transformed differences) are mainly used for measuring block distortion. Because video quality distortion and bit rate are simultaneously considered in H.264/AVC, the rate distortion optimization can also be used, as defined in equation 2.




(a) MV space (b) ZMV = (0, 0)

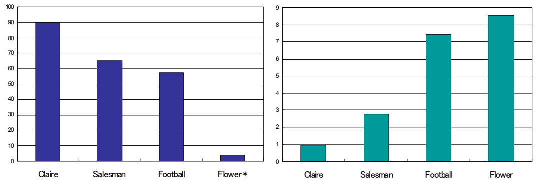
(Fig. 9) (a) A ZMV block found in MV space, (b) Location of the ZMV in search range = ±7.

$$RD\cos t = SAD + \lambda_{Mode} \times Rates \quad (2)$$

where λModeis the Lagrangian multiplier and it adjusts the tradeoff between bit rate and video distortion. Rates denotes the number of bits required for coding the difference between the candidate MV and the MV predictor.

Simulation experiments are conducted using the FS algorithm with the luminance of four CIF sequences (100 frames respectively) to analyze the correlation between the ZMV and the mean absolute error (MAE). The MAE is calculated as follows:

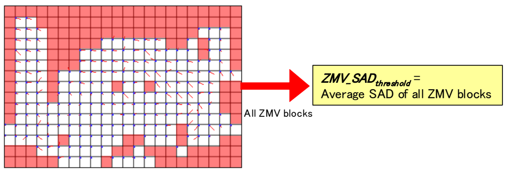
$$MAE = \frac{1}{N \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} | C_{ij} - R_{ij} | \quad (3)$$




* "Flower" MV (1, 0) = 42.60%

(a) (b)

(Fig. 10) (a) ZMV ratio comparison, (b) MAE comparison.




MV space

(Fig. 11) The existing ZMV skip method.

As a result of the simulation, Figure 10 (a) and (b) indicate that the ZMV ratio is inversely proportional to the MAE value. Based on this fact, an existing technique for effectively skipping ZMV [16] is introduced.

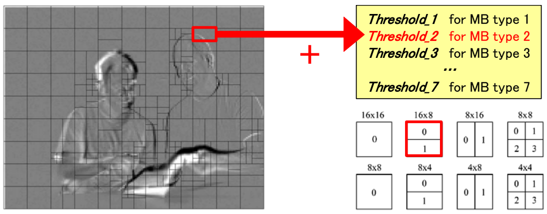
In Figure 11, the existing ZMV skip technique uses the average SAD value of all ZMV blocks as a threshold to predict ZMV. The threshold is calculated as Equation 4.

$$ZMV\_SAD_{threshold} = \frac{1}{N_{ZMV}} \sum \{SAD_{i,j} \mid MV_{i,j} = ZMV\} \quad (4)$$

Table 6 expresses the pseudo code for using the threshold in BMA. However, this technique can only be used in the motion estimation unit for MPEG-1, 2 fixed block-size.In other words, it is not appropriate for VBSME of H.264/AVC.

(Table 6) Pseudo code for using ZMV threshold in BMA.

$$Block-Matching\_Algorithm(Frame_{reference}, Frame_{current}, i, j) =$$
$$\{$$
$$\ldots$$
$$SAD_{current} \leftarrow SAD(Frame_{reference}, Frame_{current}, Block_{i,j});$$
$$if\ SAD_{current} < ZMV\_SAD_{threshold},\ return\ ZMV;$$
$$\ldots$$
$$\}$$

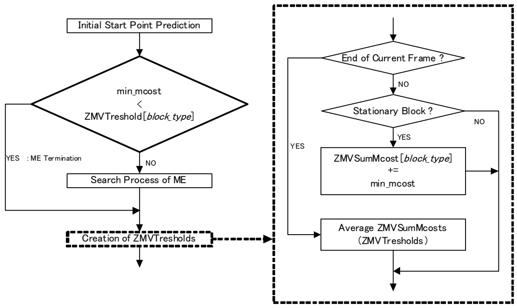



(Fig. 12) Seven variable thresholds used in the proposed ESBSM method and seven segmentations of the macroblock.

### 4.3.2 Proposed Method for VBSME

In order to implement the ZMV block skip technique in the H.264/AVC reference software JM, we propose a simple solution referred to as the efficient stationary block skip method (ESBSM). The proposed solution usesseven different variable threshold variables depending on the type of macroblock.

For example, in Figure 12, if a stationary block is a Block Type 2, its SA(T)D is added to the macroblock type 2 variable. Then the average SA(T)D is calculated right beforemoving from the current frame to the subsequent frame. The average SA(T)D can only be used as a threshold for macroblock type 2. There are seven variable thresholds in total, and each threshold value is determined according to the average SA(T)D of the corresponding macroblock type. Figure 13 displays the flow of the proposed method implemented in the H.264/AVC reference software JM. The proposed ESBSM can easily be combined with the existing ME algorithms in H.264/AVC.




(Fig. 13) Flowchart of the proposed ESBSM in the JM.

## 5. Experimental Results

The experiments are conducted at JM12.4 with the default settings of SR = 16, QP = 40, RDO= 0, and the baseline profile. Three hundred frames are used for each of the six sequences – QCIF sequences "Claire", "Container", and "Foreman", and CIF sequences "Coastguard", "Mobile", and "Stefan". In addition, 30 frames from three high-resolution HDTV sequences – "Harbour" (720p), "Ship"(2160p), and "Spinningchair" (2160p) are also used in the experiments. The experimental results, as shown in Tables 7–9, indicate that the DW+RDS algorithm combined with ESBSM (EDWS) reduces the computations of the UMHexagonS by up to 12%, while generally maintaining the PSNR.

# 6. Conclusions

In this paper, we proposed a hybrid motion estimation algorithm in H.264/AVC, which combines revised diamond search (RDS) algorithm, full diamond and dodecagon search patterns, and efficient stationary block skip method (ESBSM). In section 4.1, the proposed RDS algorithm performs faster and more exact than the original DS for CCIR601 sequences which consist of horizontal, vertical, and vigorous motion contents. We also proposed a modified UMHexagonS by using new full diamond and dodecagon search patterns. The proposed diamond web-grid search (DWS) algorithm performs a little faster than the UMHexagonS while better search quality is maintained. In the end of section 4, we suggested that an efficient stationary block skip method can be deployed for variable block-size motion estimation in H.264/AVC by using seven variable thresholds that correspond one-to-one with the variable block types of H.264/AVC. In section 5, the proposed diamond web-grid search algorithm combined with efficient stationary block skip method (EDWS) was simulated in the H.264/AVC JM software to substantiate the fact that it can further improve the speed performance of conventional motion estimation algorithms such as the UMHexagonS.

(Table 7) PSNR [dB] comparison.

| ME | | UMHexagonS | EDWS |
|---|---|---|---|
| QCIF | Claire | 31.384 | -0.037 |
| | Container | 27.841 | -0.080 |
| | Foreman | 28.137 | -0.033 |
| CIF | Coastguard | 26.663 | -0.020 |
| | Mobile | 24.342 | -0.010 |
| | Stefan | 26.339 | -0.007 |
| HDTV | Harbour | 28.294 | -0.008 |
| | Ship | 34.395 | -0.023 |
| | Spinningchair | 37.166 | -0.014 |

(Table 8) Bit rate [%] comparison.

| ME | | UMHexagonS | EDWS |
|---|---|---|---|
| QCIF | Claire | 7528 | -0.97 |
| | Container | 8414 | -1.18 |
| | Foreman | 31970 | 0.11 |
| CIF | Coastguard | 111654 | -0.28 |
| | Mobile | 229682 | -0.44 |
| | Stefan | 232526 | 0.44 |
| HDTV | Harbour | 1076424 | -0.10 |
| | Ship | 8442496 | 1.37 |
| | Spinningchair | 8048600 | 0.33 |

(Table 9) Motion estimation time [%] comparison.

| ME | | UMHexagonS | EDWS |
|---|---|---|---|
| QCIF | Claire | 8836 | -8.47 |
| | Container | 10744 | -10.41 |
| | Foreman | 13035 | -9.90 |
| CIF | Coastguard | 56593 | -12.53 |
| | Mobile | 61016 | -11.46 |
| | Stefan | 57152 | -9.99 |
| HDTV | Harbour | 34877 | -5.86 |
| | Ship | 299666 | -6.34 |
| | Spinningchair | 269397 | -5.55 |

# Acknowledgments

This research was supported in part by CREST(Core Research for Evolutional Science and Technology), JST(Science and Technology Agency).

# References

[1] K. R. Rao, J. J Hwang, "Techniques and Standards for Image, Video and Audio Coding,"

Englewood Cliffs, NJ: Prentice Hall, 1996.
[2] H. Schwarz, T. Wiegand, "The emerging JVT/H.26L video coding standard," Proc. of IBC 2002, Amsterdam, The Netherlands, Sep. 2002.
[3] "Draft ITU-T Rec. and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264-ISO/IEC 14 496-10 AVC)," Joint Video Team (JVT) of ITU-T and ISO/IEC JTC1, Geneva, JVT of ISO/IEC MPEG and ITU-T VCEG, JVT-G050r1, 2003.
[4] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, T. Ishiguro, "Motion compensated interframe coding for video conferencing," Proc. Nat. Telecommun. Conf., New Orleans, LA, Nov. 29 – Dec. 3 1981, pp. G5.3.1 – G5.3.5.
[5] R. Li, B. Zeng, M. L. Liou, "A new three-step search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol., Aug. 1994, vol. 4, pp. 438 – 442.
[6] L. M. Po, W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technol., June. 1996, vol. 6, pp. 313 – 317.
[7] L. K. Liu, E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," IEEE Trans. Circuits Syst. Video Technol., Aug. 1996, vol. 6, pp. 419 – 423.
[8] J. Y. Tham, S. Ranganath, M. Ranganath, A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol., Aug. 1998, vol. 8, pp. 369 – 377.
[9] S. Zhu, K. K. Ma, "A new diamond search algorithm for fast block matching motion estimation," IEEE Trans. Image Process., Feb. 2000, vol. 9, no. 2, pp. 287 – 290.
[10] C. Zhu, X. Lin, L. P. Chau, "Hexagon-based search pattern for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technol., May. 2002, vol. 12, pp. 349 – 355.
[11] C. H. Cheung, L. M. Po, "A novel cross-diamond search algorithm for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technol., Dec. 2002, vol. 12, no. 12, pp. 1168 – 1177.
[12] X. Jing, L. P. Chau, "An efficient three-step search algorithm for block motion estimation," IEEE Trans. Multimedia., June. 2004, vol. 6, no. 3, pp. 435 – 438.
[13] C. H. Cheung, L. M. Po, "Novel cross-diamond-hexagonal search algorithms for fast block motion estimation," IEEE Trans. Multimedia., Feb. 2005, vol. 7, no. 1, pp. 16 – 22.
[14] Z. Chen, Y. He, J. Xu, "Hybrid unsymmetrical-cross multi-hexagon-grid search strategy for integer pel motion estimation in H.264," Proc. Picture Coding Symp., Saint Malo, Apr. 2003, pp. 17 – 22.
[15] H.264/AVC Reference Software Joint Model, Download: http://iphome.hhi.de/suehring/tml/
[16] S. Hiratsuka, S. Goto, T. Ikenaga, "A locally adaptive subsampling algorithm for software based motion estimation," Proc. of ISCAS 2005, Kobe, Japan, May. 2005, pp. 2891 – 2894.

# ◐ 저 자 소 개 ◑


## 정 창 욱(Chang-Uk Jeong)

Chang-Uk Jeong is currently pursuing for the Ph.D. Degree in Graduate School of Global Information and Telecommunication at Waseda University, Japan.
He received the B.E. in computer engineering from Korea Polytechnic University in 2006.
He also received the M.E. degree in Graduate School of Information, Production and Systems, Waseda University, Japan in 2008. His research interests include video coding technology and multimedia application.
e-mail : jeongchanguk@gmail.com

## 최 진 구(Jin-Ku Choi)

Jin-Ku Choi received the Dr. Degree in electrical engineering from Waseda University, Japan in 2003. He is presently a Professor in the Department of Computer Engineering of Korea Polytechnic University.His current research interests include VLSI design and system codesign for signal processing. He is a member of IEEE and the IEICE of Japan.
e-mail : jkchey@kpu.ac.kr

## 이케나가 다케시(Takeshi Ikenaga)

Takeshi Ikenaga received his B.E. and M.E. degrees in electrical engineering and the Ph.D. degree in information & computer science from Waseda University, Tokyo, Japan, in 1988, 1990 and 2002 respectively. He joined LSI Laboratories, Nippon Telegraph and Telephone Corporation (NTT) in 1990, where he has been undertaking research on the design and test methodologies for high performance ASICs, a real-time MPEG2 encoder chipset, and a highly parallel LSI & system design for image understanding processing. He is presently an associate professor in the system LSI field of the Graduate School of Information, Production and systems, Waseda University. His current interests are application SoCs for image, security and network processing. Dr. Ikenaga is a member of the IPSJ and the IEEE. He received the IEICE Research Encouragement Award in 1992.
e-mail : ikenaga@waseda.jp

## 고토 사토시(Satoshi Goto)

Satoshi Goto was born in Hiroshima, Japan, 1945. He received the B.E. and the M.E. Degrees in Electronics and Communication Engineering from Waseda University in 1968 and 1970 respectively. He also received the Dr. of Engineering from the same University in 1981. He joined NEC Laboratories in 1970 where he worked for LSI design, Multimedia system and Software as GM and Vice President.. Since 2003, he has been Professor, at Graduate school of Information, Production and Systems of Waseda University at Kitakyushu. His main interest is now on VLSI design methodologies for multimedia and mobile applications. H has published 7 books,38 journal papers, 67 international conference papers with reviews. He served as GC of ICCAD and ASPDAC and was a board member of IEEE CAS society. He is IEEE Fellow, IEICE Fellow and Member of Academy Engineering Society of Japan.
e-mnail : goto@waseda.jp