韓國 컴퓨터情報學會 論文誌
第13卷 第7號, 2008. 12.

2008-13-7-2-3

# 개선된 크로스-납작한 육각 탐색 패턴을 이용한 고속 블록 정합 움직임 예측 알고리즘

남 현 우*

# A Fast Block Matching Motion Estimation Algorithm by using an Enhanced Cross-Flat Hexagon Search Pattern

Hyeon-Woo Nam *

## 요 약

동영상 압축을 위해서, 탐색 속도와 부호화된 비디오 화질이라는 두 가지 성능이 고려되어야 한다. 본 논문에서는, 동영상의 공간적 상관성과 움직임 벡터(MV)의 중심지향적인 특성을 이용하여 개선된 고속 블록 정합 알고리즘을 제안한다. 제안된 알고리즘은 현재 프레임의 인접 매크로 블록으로부터 초기 움직임 벡터를 예측하고 크로스 패턴과 납작한 육각 탐색 패턴을 이용하여 정확한 움직임 벡터를 결정한다.

성능 평가를 통해, 제안된 알고리즘은 탐색 속도와 부호화된 비디오 화질 측면 모두에서 비교대상인 육각 패턴 탐색 알고리즘(HEXBS)과 크로스-육각 패턴 탐색 알고리즘(CHS)에 비해 뛰어난 성능을 나타냄을 알 수 있었다. 제안된 알고리즘을 이용하여, 탐색 속도 측면에서는 약 31%의 성능 향상을 보였고, PSNR 측면에서도 약 0.5dB 향상되어 비디오 화질의 성능 향상을 나타내었다.

## Abstract

For video compression, we have to consider two performance factors that are the search speed and coded video's quality. In this paper, we propose an enhanced fast block matching algorithm using the spatial correlation of the video sequence and the center-biased characteristic of motion vectors(MV). The proposed algorithm first finds a predicted motion vector from the adjacent macro blocks of the current frame and determines an exact motion vector using the cross pattern and a flat hexagon search pattern.

From the performance evaluations, we can see that our algorithm outperforms both the hexagon-based search(HEXBS) and the cross-hexagon search(CHS) algorithms in terms of the search speed and coded video's quality. Using our algorithm, we can improve the search speed by up to 31%, and also increase the PSNR(Peak Signal Noise Ratio) by at most 0.5 dB, thereby improving the video quality.

▸ Keyword : 블록 정합(Block Matching), 탐색 패턴(Search Pattern), 움직임 예측(Motion Estimation), 움직임 벡터(Motion Vector), 인접 매크로 블록(adjacent macro blocks)

---

• 제1저자 : 남현우
• 접수일 : 2008. 10. 3, 심사일 : 2008. 10. 12, 심사완료일 : 2008. 12. 24.
* 동덕여자대학교 정보과학대학 컴퓨터학과 전임강사

# Ⅰ. Introduction

Block-based motion estimations are widely used for video compression owing to its effectiveness and simple implementation. In the block matching algorithm for the video compression, the largest impact factor is the decreasing of the inter-frame redundancy by the motion estimation(ME). Also, the search patterns of the different shapes or sizes and the distribution of motion vectors have a large impact on both the searching speed and the video quality.

The motion estimation algorithm use the characteristic that the pixels in the almost blocks have a same moving. The motion estimation divides a current frame and the previous frame of video sequence into equal-sized macro blocks(MB). And then, it determines a minimum difference measure block for each of the macro blocks of the current frame within its referenced frame[1].

The full search(FS) algorithm used in the MPEG 2 and the H.263 have many advantages that are the simple searching method and the high accuracy of the motion estimation, but also have a disadvantage that the FS have the many computation times for finding a motion vector which have the smallest difference measure value[1]. The variants of the fast block matching algorithm are studied for the overcome of the disadvantage of the FS algorithm. There are Three-Step Search(TSS)[2], Diamond Search (DS)[3-5], Hexagon-based Search(HEXBS)[6, 7], Cross Diamond Search(CDS)[8], Cross Hexagon Search(CHS)[9, 10], and so on[11-16].

Those block matching algorithm for the decrease of computation time use with the several points defined the search patterns in the search area or search without the motion information of the block. To enhance the local minimum of the block matching and the block matching entries are except some search points or the motion estimation processes to the illegal direction, several studies have been made on the prediction search algorithm with the motion information of the adjacent macro block in the same frame. In general, the video files with the motion have a small inter-frame redundancy because the very short time. The motion vector of the current macro block has the closer spatial co-relationship with the adjacent macro blocks[12-16].

In this paper, we propose an enhanced algorithm for the fast block-based motion estimation. In our method, we use with a motion information of the adjacent macro blocks and two types of search patterns that called a cross search pattern and a flat-hexagon search pattern. To show performances of the proposed algorithm, we conduct experimental evaluations over examined block matching algorithms and prediction searching algorithm. From the experimental results, we can see that our proposed algorithm outperforms others in terms of the search speed, and preserves the high image quality of video files being coded.

In the following section, the search pattern and the algorithm of the HEXBS and CHS will be examined to investigate its advantages. Section 3 first explains the prediction information of current macro block using the motion vector of three adjacent macro blocks, and then an explanation about the proposed cross-flat-hexagon search pattern and its algorithm follows. In Section 4, experimental results are presented of our proposed algorithm compared with the FS, DS, HEXBS, and CHS. Section 5 concludes the paper.

# Ⅱ. Related Studies

In general, a video stream has a large amount of temporal redundancy between its successive image frames, and thus the concept of the block matching ME capitalizes on such temporal redundancy.

The block matching is a procedure to search for a best match within the reference frame with respect to an image block of the corresponding target frame. The best match for an image block in a target frame

표 1. 움직임 벡터의 분포 밀도 : 그림 1의 구체적 데이터(%)
Table 1. Distribution density of motion vectors: detailed data of Fig. 1(in percentage)

| p | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| -7 | 0.03 | 0.03 | 0.03 | 0.04 | 0.05 | 0.05 | 0.06 | 0.11 | 0.05 | 0.03 | 0.03 | 0.03 | 0.04 | 0.03 | 0.04 |
| -6 | 0.02 | 0.03 | 0.04 | 0.03 | 0.04 | 0.05 | 0.05 | 0.10 | 0.05 | 0.04 | 0.04 | 0.03 | 0.03 | 0.04 | 0.03 |
| -5 | 0.03 | 0.03 | 0.03 | 0.05 | 0.04 | 0.06 | 0.07 | 0.11 | 0.05 | 0.04 | 0.06 | 0.03 | 0.03 | 0.03 | 0.02 |
| -4 | 0.04 | 0.04 | 0.04 | 0.04 | 0.05 | 0.05 | 0.08 | 0.13 | 0.07 | 0.05 | 0.04 | 0.05 | 0.03 | 0.03 | 0.03 |
| -3 | 0.04 | 0.04 | 0.04 | 0.06 | 0.05 | 0.08 | 0.10 | 0.16 | 0.08 | 0.07 | 0.04 | 0.05 | 0.04 | 0.04 | 0.03 |
| -2 | 0.06 | 0.05 | 0.07 | 0.07 | 0.07 | 0.14 | 0.16 | **0.28** | 0.14 | 0.07 | 0.06 | 0.06 | 0.05 | 0.06 | 0.05 |
| -1 | 0.19 | 0.09 | 0.13 | 0.15 | 0.22 | 0.26 | 0.50 | **0.95** | 0.33 | 0.20 | 0.19 | 0.16 | 0.13 | 0.13 | 0.04 |
| 0 | 1.24 | 0.70 | 0.99 | 0.91 | 3.13 | **4.53** | **11.29** | **45.68** | **2.35** | **1.27** | 1.51 | 1.88 | 2.27 | 0.39 | 0.38 |
| 1 | 0.37 | 0.32 | 0.94 | 0.52 | 0.44 | 0.45 | 0.69 | **2.17** | 0.60 | 0.18 | 0.10 | 0.06 | 0.05 | 0.06 | 0.09 |
| 2 | 0.16 | 0.11 | 0.11 | 0.12 | 0.13 | 0.22 | 0.22 | **0.35** | 0.15 | 0.12 | 0.08 | 0.05 | 0.05 | 0.05 | 0.03 |
| 3 | 0.05 | 0.05 | 0.05 | 0.06 | 0.07 | 0.11 | 0.12 | 0.19 | 0.10 | 0.09 | 0.07 | 0.05 | 0.04 | 0.03 | 0.03 |
| 4 | 0.03 | 0.04 | 0.04 | 0.04 | 0.08 | 0.09 | 0.08 | 0.18 | 0.06 | 0.06 | 0.05 | 0.03 | 0.03 | 0.02 | 0.03 |
| 5 | 0.03 | 0.03 | 0.03 | 0.05 | 0.06 | 0.06 | 0.06 | 0.13 | 0.06 | 0.06 | 0.06 | 0.04 | 0.03 | 0.02 | 0.03 |
| 6 | 0.02 | 0.03 | 0.03 | 0.03 | 0.04 | 0.04 | 0.05 | 0.09 | 0.05 | 0.05 | 0.04 | 0.02 | 0.03 | 0.03 | 0.03 |
| 7 | 0.04 | 0.04 | 0.03 | 0.03 | 0.03 | 0.03 | 0.05 | 0.13 | 0.05 | 0.04 | 0.03 | 0.04 | 0.03 | 0.03 | 0.03 |

is the image block that exists in the associated reference frame and has the minimum block distort measure(BDM) against block. The position of the best match is expressed with a motion vector, which is in a form (x, y) and integers x and y represent the distance of the best match from the left-top point of block. By coding such motion vectors between reference frames and target frames, we can reduce the data size of video files[1].

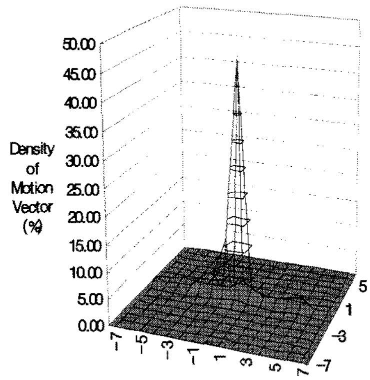
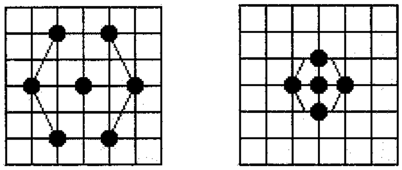



그림 1. 실험 동영상의 움직임 벡터 밀도
Fig 1. Motion vector density of experimental videos

Figure 1 shows the distribution densities of motion vectors of six popular sample video files. In this figure, the peak point at (0,0) expresses the fact that the motion vector of (0,0) has the largest occurrence density. Here, motion vector (0,0) means that the best match has the same frame coordinates with respect to its target image block.

Table. 1 gives the detailed data regarding the motion vector density of Figure 1. In this table, a figure in each cell is the distribution density value of the corresponding motion vector. Therefore, we can see that in the table more than 45 percent of motion vectors have (0, 0) as their values. Moreover, more than 68 percent of motion vectors belong to the motion vector sets of (0,0), $(0,\pm1)$, $(\pm1,0)$, $(0,\pm2)$, and $(\pm2,0)$.

Figure 2 shows the patterns of the HEXBS algorithm. The HEXBS algorithm uses a large hexagon-shaped pattern(LHSP) and a small hexagon-shaped pattern(SHSP). Figure 2(a) shows a large hexagon-shaped search pattern that consists of seven checking points with the center surrounded by six points of the hexagon with distance 2 and $\sqrt{5}$ from the center point, respectively[6, 7].



(a) 큰 육각패턴　　(b) 작은 육각패턴
(a) LHSP　　(b) SHSP

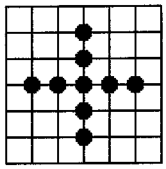
그림 2. 육각패턴 탐색 알고리즘에 적용된 두 가지 탐색 패턴
Fig 2. Two search patterns are employed in the HEXBS algorithm

From the figure 2(a), we can see the six points are approximately uniformly distributed around the center. In the search process using the HEXBS algorithm, the large hexagon- shaped search pattern keeps advancing with the center moving to any of the six points. Whichever point the center of the search pattern moves to, there are always three new points emerging, and the other three points are being overlapped. Figure 2(b) shows a smaller shrunk hexagonal pattern covering four checking points (left, right, up, and down dots around the center with distance 1) in the motion field, which is finally used in the focused inner search[6, 7].

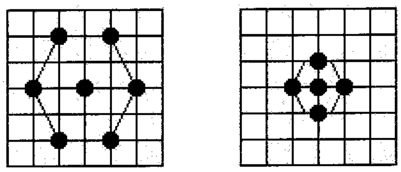
Figure 3 shows the patterns of the CHS algorithm. The CHS algorithm uses an initial cross search pattern(CSP) and a large hexagon-shaped pattern and a small hexagon-shaped pattern[9, 10].

The CHS algorithm differs from the HEXBS algorithm by performing a CSP in the first step and employing a halfway-stop technique for quasi-stationary or stationary candidate blocks. Below summarizes[9, 10].

In the starting step, a minimum BDM is found from the nine search points of the CSP located at the center of search window. If the minimum BDM point occurs at the center of the CSP, the search stops. Otherwise, two additional search points closest to the current minimum of the central CSP are checked, i.e., two of the four candidate points located at $(\pm1,\pm1)$. If the minimum BDM found in previous step located at the middle wing of the CSP, i.e., $(\pm1,0)$ or $(0,\pm1)$, and the new minimum BDM found in this step still coincides with this point, the search stops. Otherwise, a new LHSP is formed by repositioning the minimum BDM found in previous step, and then perform same step of the HEXBS algorithm until the final solution is found[9, 10].



(a) 크로스 패턴
(a) CSP



(b) 큰 육각패턴　　(c) 작은 육각패턴
(b) LHSP　　(c) SHSP

그림 3. 크로스-육각패턴 탐색 알고리즘에 적용된 세 가지 탐색 패턴
Fig 3. Three search patterns are employed in the CHS algorithm

## III. Proposed Algorithm

In order to obtain an accurate MV predicted of the current block, two factors need to be considered the choice of the neighboring blocks whose MV will be used to calculate the predicted MV, and the algorithm used for computing the predicted MV[13, 14, 16].

In the spatial domain, since all the blocks within a video frame are processed in a raster-scan order, the adjacent macro blocks whose MV are available for reference are on the immediate left, above and above-right to the current block, as shown in Figure 4[10, 13, 14, 16].

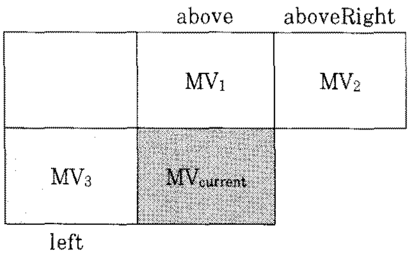



그림 4. 참조되는 인접 매크로 블록
Fig 4. Referenced the Adjacent Macro Blocks

The blocks in other nearby positions are less correlated to the current block and thus are not reliable for prediction. This way is the prediction adopted in some international standards such as H.263 for differential coding of MV[10, 13, 14].

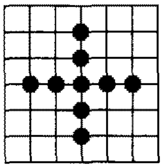
If the motion vectors of the adjacent macro blocks are $MV_0, MV_1$ and $MV_2$, the predicted motion vector $MV_{current}$ is the median of the each motion vector of the adjacent macro block, as equation (3.1)[10, 13, 14].

$$MV_{current} = median(MV_0, MV_1, MV_2) \quad \cdots\cdots\cdots\cdots (3.1)$$

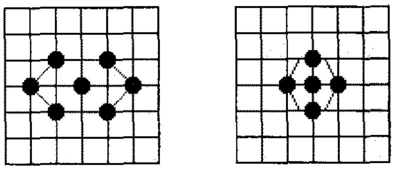
If MV prediction is determined, generally the search pattern use for the block matching. The typical of the search pattern is the TSS algorithm or the DS algorithm. These predicted search algorithms evaluated the excellent performance in the speed and the coded video quality. However, when the co-relation of the motion vector is loose, the video quality can reduce by the illegal motion estimation[10, 13, 14].

Our experimental results show that these predicted blocks and prediction criteria yield fairly similar performance in terms of MAD and PSNR.

Our new search patterns are shown in Figure 5. The search pattern of Figure 5(a) is called the cross search pattern(CSP) used to check the nine points of (0,0), (0,±1), (±1,0), (0,±2), and (±2,0). The pattern of Figure 5(b) is the flat-hexagon search pattern covering the six points of (±1,1), (±1,−1), and (±2,0). With the combination of the two search patterns, we detect the best match in a rapid speed.



(a) 크로스 패턴
(a) CSP



(b) 큰 납작한 육각패턴 (c)작은 납작한 육각패턴
(b) FHSP (c) SHSP
그림 5. 제안된 알고리즘의 탐색 패턴
Fig 5. Search patterns used in the proposed algorithm

Moreover, the flat-hexagon pattern of Figure 5(b) can provide a better performance while coding video frames that contain frequently changing scenes or images including quickly moving objects. And, the search pattern of Figure 5(c) is called the small hexagon-shaped pattern. This pattern is used to decide a best motion vector at the last search step.

Below summarizes the proposed enhanced cross-flat-hexagon search algorithm.

Step 1 (Determining): If a searching start point is found from the median of the motion vectors the three adjacent macro blocks, then go to Step 2.
Step 2 (Starting): A minimum BDM is found from the nine search points of the CSP located at the predicted searching start point. If the minimum BDM point occurs at the center of the CSP, the search stops. This is called the second step stop. Otherwise, go to Step 3.

Step 3 (Small-hexagon searching): Two additional search points of the FHSP closest to the current minimum of the central CSP are checked, i.e., two of the four candidate points located at$(\pm1,\pm1)$. If the minimum BDM found in previous step located at the middle wing of the CSP, i.e., $(\pm1,0)$ or $(0,\pm1)$, and the new minimum BDM found in this step still coincides with this point, the search stops. This is the called the third step stop. Otherwise, go to Step 4.

Step 4 (Flat-hexagon searching): A new FHSP is formed by repositioning the minimum BDM found in previous step as the center of the FHSP. If the new minimum BDM point is still at the center of the newly formed FHSP, then go to Step 5; otherwise, this step is repeated again.

Step 5 (Ending): With the minimum BDM point in the previous step as the center, a new SHSP is formed. Identify the new minimum BDM point from the four new candidate points, which is the final solution for the motion vector.

## IV. Performance Evaluations

### 4.1 Experiment Environments

For the performance evaluation, we use two CIF video files (352 by 288 in pixels), two SIF video files (352 by 240 in pixels), and two QCIF video files (176 by 144 in pixels).

To show our performance advantages, we conduct several experimental comparisons with other algorithms such as a FS, DS, HEXBS, and CHS, and in terms of the search speed and coded video's quality. To run the block match algorithms to be evaluated, we use a PC server with a Pentium 4 CPU and 512 MB of memory. In all experiments, block size of 16, and search window w=±7 are used.

As the measure of the BDM, we use the SAD (sum of absolute difference), as equation (4.2)[1, 2, 9-11].

$$SAD = \sum_{x=1}^{MB}\sum_{y=1}^{MB}\left|I_t(x,y) - I_{t-1}(x+dx, y+dy)\right| \quad \cdots\cdots\cdots (4.1)$$

In the above equation (4.1), notation MB is the size of an image block to be coded. The notation $I_t(x, y)$ represents the color of a pixel at (x, y) in the target frame, while $I_{t-1}(x, y)$ is the color of the pixel in the reference frame. Here, *(dx, dy)* represents the distance between the reference image block and the target image block. In the algorithm, if a best match for the target block has the same coordinates as those of the target block, then the motion vector (0, 0) is returned[1, 2, 9-11].

As measures of coded video's quality, we employ both of popular ones of MAD(Mean Absolute Difference) and PSNR[9-11].

How to obtain MAD and PSNR values is given in equations (4.2) and (4.3), respectively.

$$MAD = \left(\frac{1}{M \times N}\right)\sum_{x=1}^{M}\sum_{y=1}^{N}\left|O_t(x,y) - E_t(x,y)\right| \quad \cdots\cdots\cdots (4.2)$$

$$PSNR = 10\log_{10}\frac{255^2}{MSE}$$

where,

$$MSE = \left(\frac{1}{M \times N}\right)\sum_{x=1}^{M}\sum_{y=1}^{N}\left|O_t(x,y) - E_t(x,y)\right|^2 \quad \cdots\cdots (4.3)$$

In the formulas above, notations $M$ and $N$ represent the sizes of video frames in horizontal axis and vertical axis, respectively. $O_t(x, y)$ represents the color of a pixel at (x, y) in reference frame, while $E_t(x, y)$ is the color of the corresponding pixel in the target frame. According to the definitions of MAD and PSNR, a coded video yielding less MAD and PSNR is said to have a better image quality[9, 10, 11].

Throughout our experiments, we will show that our proposed algorithm provides a faster block search speed than other algorithms, while maintaining the high image quality by yielding less MAD and PSNR.

## 4.2 Experiment Results

Table 2 shows the experimental results concerning the search speed of the block matching algorithms. In the table, our proposed algorithm is denoted by ECFHS(enhanced cross-flat-hexagon search algorithm) for short. In this Table 2, the columns of PN are the numbers of points on which the BDM needs to be computed for detecting a best motion vector. The SRATE columns give the improvement rates of each algorithm's search speeds with respect to FS.

As known from Table 2, compared with the HEXBS and the CHS algorithm, our proposed algorithm provides the best searching speed among the experimented algorithms. This efficiency of block searching mainly comes from the use of our cross-flat-hexagon search pattern.

Since the fast search speed could adversely affect the coded image quality, it is important to prevent a

표 2. 탐색 속도의 비교
Table 2. Comparisons of search speeds.

| VIDEO NAME | RESULT ITEM | BMA | | | | |
|---|---|---|---|---|---|---|
| | | FS | DS | HEXBS | CHS | ECFHS |
| Akiyo (CIF) | PN | 204.28 | 12.29 | 10.35 | 8.89 | 8.67 |
| | SRATE | 1.00 | 16.62 | 19.74 | 22.99 | 23.55 |
| Stefan (CIF) | PN | 204.28 | 16.77 | 12.98 | 10.86 | 10.38 |
| | SRATE | 1.00 | 12.18 | 15.74 | 18.81 | 19.68 |
| Flower Garden (SIF) | PN | 202.05 | 16.81 | 13.15 | 11.98 | 10.02 |
| | SRATE | 1.00 | 12.02 | 15.37 | 16.87 | 20.16 |
| Popp (SIF) | PN | 202.05 | 15.39 | 12.27 | 11.89 | 11.86 |
| | SRATE | 1.00 | 13.13 | 16.47 | 16.99 | 17.03 |
| Akiyo (QCIF) | PN | 184.56 | 11.44 | 9.65 | 8.28 | 8.21 |
| | SRATE | 1.00 | 16.14 | 19.13 | 22.29 | 22.47 |
| Table Tennis (QCIF) | PN | 184.56 | 13.88 | 11.22 | 11.43 | 10.73 |
| | SRATE | 1.00 | 13.29 | 16.44 | 16.15 | 17.19 |

표 3. MAD 와 PSNR 측면에서의 부호화된 비디오 화질
Table 3. Coded video's qualities in terms of MAD and PSNR

| VIDEO NAME | RESULT ITEM | BMA | | | | |
|---|---|---|---|---|---|---|
| | | FS | DS | HEXBS | CHS | ECFHS |
| Akiyo (CIF) | MAD | 0.603 | 0.604 | 0.617 | 0.605 | 0.604 |
| | PSNR | 42.780 | 42.760 | 42.498 | 42.682 | 42.739 |
| Stefan (CIF) | MAD | 6.715 | 7.717 | 7.929 | 7.883 | 7.111 |
| | PSNR | 25.693 | 24.655 | 24.501 | 24.526 | 24.816 |
| Flower Garden (SIF) | MAD | 8.520 | 8.768 | 9.509 | 8.822 | 8.654 |
| | PSNR | 23.994 | 23.806 | 23.320 | 23.703 | 23.792 |
| Popp (SIF) | MAD | 4.784 | 4.900 | 5.080 | 4.958 | 4.908 |
| | PSNR | 27.077 | 26.780 | 26.379 | 26.685 | 26.846 |
| Akiyo (QCIF) | MAD | 0.488 | 0.488 | 0.491 | 0.488 | 0.488 |
| | PSNR | 43.851 | 43.851 | 43.813 | 43.851 | 43.851 |
| Table Tennis (QCIF) | MAD | 4.585 | 4.738 | 5.007 | 4.773 | 4.741 |
| | PSNR | 28.197 | 27.686 | 27.259 | 27.590 | 27.744 |

faster searching procedure from deteriorating the image quality of coded video files. And, to show the image quality preserving in our algorithm, we give Table 3.

The columns of MAD and PSNR in the Table 3 correspond to the image quality measures of equation (4.1) and (4.2), respectively.

From the experimental results, the proposed algorithm provides the equivalent or the exceeded image quality, compared with other algorithms.

Figure 6 shows the proposed algorithm efficiently runs for video files with much difference between successive frames, as well as other video files with less inter-frame difference.

In Figure 6(a), the proposed ECFHS algorithms provides a faster search time than the HEXBS and CHS algorithm in the case of video "Stefan(CIF Sequence)" which does include quick changes of scenes and thus has much inter-frame difference. And, the experiment in Figure 6(b), the proposed ECFHS algorithm provides a faster search speed than the HEXBS and CHS algorithm in the case of "Flower Garden(SIF Sequence)" which does also include quick changes of scenes and thus has much inter-frame difference. In addition, in the case of smooth video, the proposed ECFHS algorithms provides a faster search time than the HEXBS and CHS algorithm.

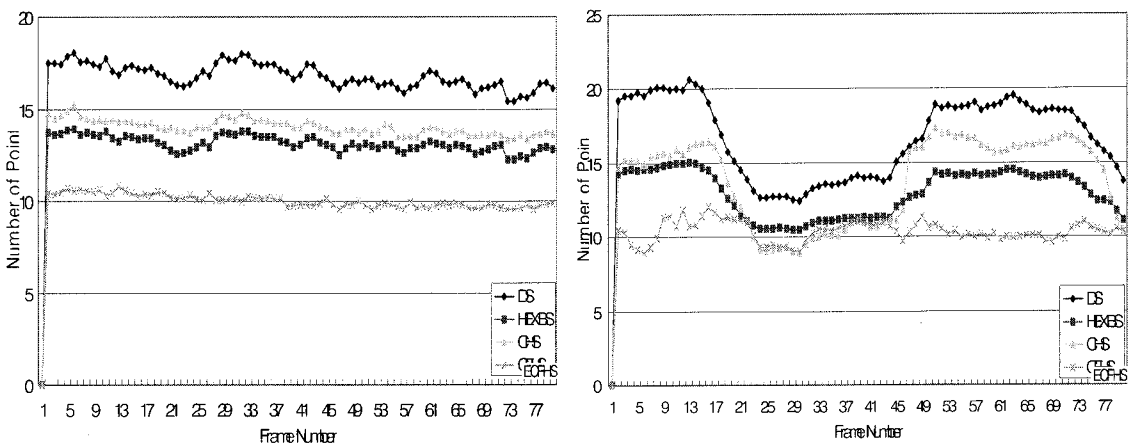
Therefore, the proposed algorithm provides a better search performance for the most video, being invulnerable to the degree of temporal redundancy.

## V. Conclusions

For the development of an efficient algorithm for block matching motion estimation, we have to consider two major performance factors, that is, the search speed and coded video's quality. Since there is often a trade-off between these two performance factors, it is not easy to devise any algorithm that provides the best performance figures in all these aspects.

The HEXBS and CHS have advantages such as a very high search speed and preserving of coded video's image quality, compared with the naive full search algorithm. Although theses algorithms have such desirable characteristics, their performance advantages are rather susceptible to inter-frame redundancy of compressed video.

To prevent such shortcomings, we proposed new fast block matching algorithms that use the motion




(a) "Stefan(CIF)" 영상의 BDM 계산
(a) BDM computation for "Stefan (CIF)"

(b) "Flower Garden (SIF)" 영상의 BDM 계산
(b) BDM computation for "Flower Garden (SIF)"

그림 6. 실험 동영상에 대한 BDM 계산 비교
Fig 6. Comparisons of the BDM computation numbers with respect to the experimental video

information and a cross-flat-hexagon search pattern. The motion information of the adjacent macro blocks used to overcome of the lower video quality with the predict information and a cross-flat-hexagon search pattern used to search quickly the exact motion vectors with the cross search pattern and a flat-hexagon search pattern.

From the performance experiments, we can see that our algorithm provides a better performance in terms of the search speed and the coded video's quality, compared with both the HEXBS and CHS algorithms. In particular, our algorithm provides performance advantages regardless of the amounts of temporal redundancy between video frames. Using our algorithm, we can improve the search speed by up to 31%, and also increase the PSNR(Peak Signal Noise Ratio) by at most 0.5 dB, thereby improving the video quality.

## References


[1] Information Technology-Generic Coding of Moving Pictures and Associated Audio Information: Video, ISO/IEC 13818-2(MPEG-2 Video), 2000.

[2] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated Interframe Coding for Video Conferencing", In Proc. of the International Telecommunications Conference, New Orleans, LA, pp. G5.3.1-G5.3.5, 1981.

[3] Shan Zhu, and Kai-Kuang Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Transactions on Image Processing, Vol. 9, No. 2, pp. 287-290, 2000.

[4] T. Sappasitwong, S. Aramvith, S. Jitapunkul, A. Tamtrakarn, P. Kitti-punyangam, and H. Kortrakulkij, "Adaptive asymmetric diamond search algorithm for block-based motion estimation", In Proc. of the International Conference on Digital Signal Processing, pp. 563-566, 2002.

[5] Sung-Yong Hong, "Fast Block-Matching Motion Estimation Using Constrained Diamond Search Algorithm", Journal of The Korea Society of Computer and Information, Vol. 8, No. 4, pp. 13-20, 2003.

[6] Ce Zhu, Xiao Lin, Lap-Pui Chau, Keng-Pang Lim, Hock-Ann Ang, and Choo-Yin Ong, "A novel hexagon-based search algorithm for fast block motion estimation", In Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 1593-1596, 2001.

[7] Ce Zhu, Xiao Lin, and Lap-Pui Chau, "Hexagon-Based Search Pattern for Fast Block Motion Estimation", IEEE Transactions on Circuits & System for Video Technology, Vol. 12, No. 5, pp. 349-355, 2002.

[8] C. H. Cheung, and L. M. Po, "A Novel Cross-Diamond Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Transactions on Circuits & System for Video Technology, Vol. 12, No. 12, pp. 1168-1177, 2002.

[9] In-Young Park, Hyeon-Woo Nam, Youngcheul Wee, and Ha-Jine Kimn, "A New Cross and Hexagonal Search Algorithm for Fast Block Matching Motion Estimation", The KIPS Transactions : Part B, Vol. 10-B, No. 7, pp. 811-814, 2003.

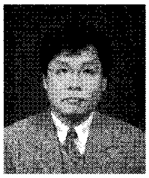
[10] Hyeon-Woo Nam, "A Fast Block Matching Motion Estimation Algorithm by using the Enhanced Cross-Hexagonal Search Pattern", Journal of The Korea Society of Computer and Information, Vol. 11, No. 4, pp. 77-85, 2006.

[11] Hyeon-Woo Nam, "A Flat Hexagon-based Search Algorithm for Fast Block Matching Motion Estimation", Journal of The Korea Society of Computer and Information, Vol. 12, No. 1, pp. 77-85, 2007.

[12] Oscal T.-C. Chen, "Motion Estimation Using a One-Dimensional Gradient Descent Search", IEEE Transactions on Circuits & System for Video Technology, Vol. 10, No. 4, pp. 608-616, 2000.

[13] A. M. Tourapis, O. C. Au, and M. L. Liou, "Highly efficient predictive zonal algorithms for fast block-matching motion estimation", IEEE Transactions on Circuits & System for Video Technology, Vol. 12, No. 10, pp. 934-947, 2002.
[14] F. Moschetti, M. Kunt, and E. Debes, "A Statistical Block-Matching Motion Estimation", IEEE Transactions on Circuits & System for Video Technology, Vol. 13, No. 4, pp. 417-431, 2003.
[15] KIbeom Kim, Chan-Young Jeong, and Min-Cheol Hong, "Fast Motion Estimation Using Local Statistics of Neighboring Motion Vectors", Journal of Broadcating Engineering, Vol. 13, No. 1, pp. 128-136, 2008.
[16] Jung-Jun Kim, Gwang-gil Jeon, and Je-chang Jeong, "Adaptive Selection of Fast Block Matching Algorithms for Efficient Motion Estimation", Journal of Korea Information and Communication Society, Vol. 33, No. 1, pp. 19-33, 2008.

## 저 자 소 개



**남 현 우**

1993년 아주대학교 공과대학 전자계산학과 (공학사)
1996년 아주대학교 대학원 컴퓨터공학과(공학석사)
2004년 아주대학교 대학원 컴퓨터공학과(공학 박사)
2005년~현재: 동덕여자대학교 컴퓨터학과 전임강사
관심분야 : 컴퓨터그래픽스, 영상처리, 멀티미디어 및 컴퓨터 게임