

특집 논문-07-12-4-01

## 움직임 벡터의 시공간적 상관도에 따른 효율적인 움직임 추정 기법

최민석<sup>a)\*</sup>, 김종호<sup>a)</sup>, 정제창<sup>a)</sup>

## An Efficient Motion Estimation Technique using the Spatial and Temporal Correlations

Minseok Choi<sup>a)\*</sup>, Jongho Kim<sup>a)</sup>, and Jechang Jeong<sup>a)</sup>

## 요약

비디오 압축 기법에서 움직임 추정(Motion Estimation)은 매우 중요한 부분을 차지하는데, 그것은 움직임 추정이 화질과 인코딩 시간에 직접적으로 영향을 미치기 때문이다. 가장 기본적인 움직임 추정 기법은 전역 탐색 기법(Full Search)인데, 이는 가장 좋은 화질을 보여주는 것 하지만 매우 많은 계산량이 필요하다는 단점이 있다. 따라서 좋은 화질을 유지하면서도 계산량을 낮추기 위한 많은 고속 알고리즘들이 제안되었다. 그 중 PMVFAST는 움직임 벡터 간의 연관성을 이용하여 계산량을 낮추면서도 전역 탐색 기법에 거의 근접한 화질을 얻어낼 수 있었다. 이 논문에서는 현재 프레임과 이전 프레임의 움직임 벡터에 기초하여 현재 매크로블록의 움직임 벡터를 예측하는 새로운 예측 방법을 제시한다. 실험결과에 따르면 제안한 알고리즘은 PMVFAST보다 빠르면서도 전역 탐색기법보다도 높은 평균 PSNR을 보여주었다.

## Abstraction

Motion Estimation (ME) is a core part of most video compression systems since it affects directly the output video quality and the encoding time. The most basic method of ME, Full Search (FS) gives the highest visual quality but also has the problem of significant computational load. To solve this problem, many fast algorithm has been proposed. Among them, MVFAST and PMVFAST show impressive results in video quality and the computational load by using the correlation between motion vectors of adjacent blocks. In particular, PMVFAST reduces search points dramatically and also gives very high video quality by using the median predictor. In this paper, we propose a new algorithm that uses the redefined median predictor which reduces the number of search points and yields a high visual quality by reducing the number of thresholds and early termination conditions.

Keyword : Motion Estimation, MVFAST, PMVFAST, median prediction

## 1. 서론

대부분의 비디오 압축 표준들은 프레임 간의 시간적 중

복성을 줄이기 위해 블록 정합 알고리즘(Block Matching Algorithm)을 사용한다. 블록 정합 알고리즘에서 각 프레임은 겹치지 않는 여러 개의 매크로블록으로 분할되고, 부호화하고자 하는 매크로블록은 현재 매크로블록으로부터 참조 프레임에서 현재 매크로블록과 가장 유사한 블록으로의 움직임 벡터로 대체된다. 어느 블록이 현재 프레임과 가장 유사한지를 측정하는 기준은 여러 가지가 있는데, 그 중 가장 많이 사용되는 것은 SAD(Sum of Absolute Difference)

a) 한양대학교 전자통신컴퓨터공학과

Department of Electronics and Computer Engineering, Hanyang University

\* 교신저자 : 최민석(naranicca@gmail.com)

\*\* 이 논문은 IWAIT2007에 발표된 논문임. 본 연구는 서울시 산학연협력 사업으로 구축된 서울 미래형콘텐츠컨버전스 클러스터 지원으로 수행되었습니다.

$$SAD(v_x, v_y) = \sum_{m=0n=0}^{M-1N-1} |F_t(x+m, y+n) - F_{t-i}(x+v_x+m, y+v_y+n)| \tag{1}$$

이다. 현재 프레임을  $M \times N$  크기로 분할하였을 때  $(x, y)$ 에 위치한 현재 매크로블록과  $(x+v_x, y+v_y)$ 에 위치한 블록 간의 SAD는 다음과 같은 식으로 표현할 수 있다.

여기에서  $F_t$ 는 현재 프레임이고,  $F_{t-i}$ 는 참조 프레임이다. 식을 통해 알 수 있듯이, SAD를 한 번 구하기 위해서는  $M \times N$  번의 뺄셈이 필요하다. 만약 탐색 범위가  $S_x \times S_y$ 일 때 전역 탐색 기법을 사용한다면 매크로블록 한 개당  $S_x \times S_y \times M \times N$  번의 뺄셈이 필요한데, 이는 실로 엄청난 계산량이라고 할 수 있다.

이러한 계산량을 줄이기 위해서 많은 고속 알고리즘들이 제안되어 왔는데 2D log 탐색<sup>[4]</sup>, TSS (Three-Step Search)<sup>[5]</sup>, NTSS (New Three-Step Search)<sup>[6]</sup>, 다이아몬드 탐색 기법<sup>[7,8]</sup> 등이 그것이다. 이러한 알고리즘들은 주어진 탐색 패턴에 따라 SAD 계산 횟수를 줄임으로써 계산량을 줄였지만, 그에 따라 화질이 열화되는 문제점이 있었다. 하지만 최근에 제안된 MVFAST (Motion Vector Field Adaptive Search Technique)나 PMVFAST (Predictive Motion Vector Field Adaptive Search Technique)는 움직임 벡터 간의 상관관계를 이용하여 계산량과 화질의 열화를 획기적으로 줄이는데 성공하였다<sup>[9-11]</sup>. 특히 PMVFAST의 경우 많은 알고리즘들이  $(0, 0)$  벡터를 중심으로 탐색을 하는데 비해 주변 매크로블록의 모션 벡터들의 중간값을 중심으로 탐색을 실시하여 빠르면서도 높은 화질을 보여주었으며 영상에 따라 전역 탐색 기법보다 높은 PSNR을 얻을 수도 있는데 이는 움직임 벡터 간의 상관성이 높아지면서 움직임 벡터를 부호화하는데 필요한

비트가 줄어들기 때문에 가능한 현상이다.

이 논문에서는 PMVFAST보다 더 정확히 움직임 벡터를 예측하고 움직임 벡터 간의 상관성을 더욱 높이는 새로운 움직임 추정 기법을 제안한다. PMVFAST의 경우 주변 블록의 움직임 벡터를 이용하여 탐색 시작점을 결정하는데 비해, 제안하는 알고리즘은 주변 블록 뿐만 아니라 참조 프레임에서 부호화하고자 하는 매크로블록과 같은 위치에 있는 매크로블록의 움직임 벡터도 이용하여 예측을 하기 때문에 보다 정확한 시작점 예측이 가능하다. 시작점 예측이 비교적 정확하게 이루어지기 때문에 작은 탐색 패턴만을 이용하여 탐색점의 수를 더 줄일 수 있다. 탐색점의 수를 줄임으로써 종료조건 등을 간소화 할 수 있고 이는 더 높은 PSNR을 보장할 수 있게 해 준다.

2장에서는 기존에 제안된 알고리즘에 대해 간략히 알아보고, 3장에서는 제안하는 알고리즘을 설명할 것이다. 4장에서는 실험 결과를 통해 이 알고리즘의 효율성을 살펴본다.

## II. 기존의 고속 알고리즘

대부분의 고속 움직임 추정 알고리즘은 GDS (Gradient Descent Search) 방법을 사용한다. GDS 방법은 일종의 탐욕적 알고리즘(Greedy Algorithm)인데, 탐색 초기점을 결정하고 다양한 탐색 패턴을 이용하여 반복적으로 최소의 SAD를 찾아나가는 방법이다.

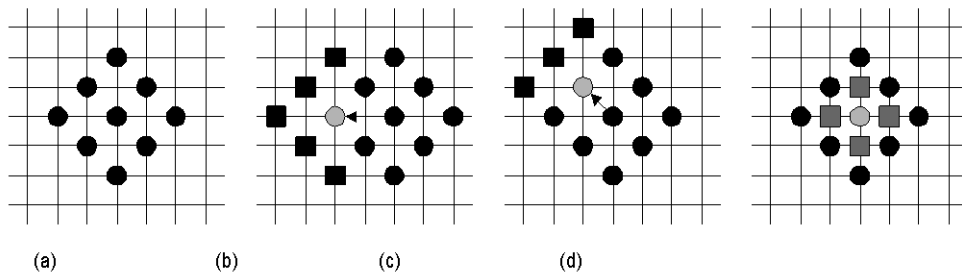


그림 1. 다이아몬드 탐색에서의 LDSP와 SDSP. (a)-(c) LDSP, (d) SDSP  
 Fig. 1. LDSP and SDSP in Diamond Search. (a)-(c) LDSP, and (d) SDSP

초창기의 고속 알고리즘들은 탐색 초기점을 (0, 0)으로 결정하는 경우가 많았는데, 이는 많은 매크로블록의 움직임 벡터가 (0, 0) 주변이라는 통계를 이용한 것이다. 이에 속하는 알고리즘 중 다이아몬드 탐색 기법이 비교적 좋은 성능을 보여 주었는데, 다이아몬드 탐색은 LDSP (Large Diamond Search Pattern)과 SDSP (Small Diamond Search Pattern)이라는 2가지 탐색 패턴을 이용하였다.

다이아몬드 탐색 기법은 배경과 같이 거의 움직임이 없는 부분에서는 좋은 결과를 보여주지만 움직임이 큰 부분에서는 좋은 성능을 보여주지 못한다. 이러한 단점을 보완한 알고리즘이 MVFAST와 PMVFAST이다. MVFAST와 PMVFAST는 현재 매크로블록의 움직임 벡터가 주변 매크로블록의 움직임과 상관성이 높다는 점을 이용하여 탐색 시작점을 결정한다.

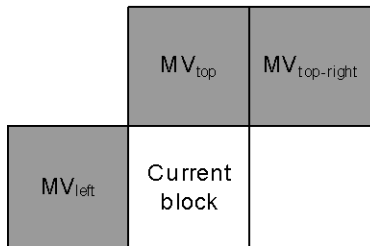


그림 2. MVFAST와 PMVFAST에서 탐색 시작점을 예측하기 위해 사용하는 예측 벡터

Fig. 2. Motion vectors used to predict the starting point in MVFAST and PMVFAST.

특히 PMVFAST의 경우 그림 2의 3개의 예측 벡터들의 중간값으로 탐색 시작점을 예측함으로써 매우 정확한 시작점 예측이 가능하고, 움직임 벡터를 부호화하는데 필요한 비트를 줄임으로써 화질을 높이는 데도 성공하였다.

### III. 제안하는 고속 움직임 추정 알고리즘

제안하는 알고리즘은 PMVFAST에 기초하고 있다. 하지만 속도와 화질에서 뚜렷한 개선을 얻기 위하여 새로운 시작점 예측 방법을 적용하였다. 시작점을 더욱 정확하게 예측할 수 있게 되면 속도 면에서 뚜렷한 효과를 얻을 수 있을

뿐 아니라, 문턱값과 조기 종료 조건을 줄일 수 있어 높은 화질도 이끌어 낼 수 있기 때문이다.

#### 1. 새로운 중간값 예측

PMVFAST는 시작점을 결정하기 위해 3개의 움직임 벡터를 사용한다. 시작점이 결정된 후에는 영벡터와  $MV_{t-1}$ 이 추가된다. 표 1은 PMVFAST를 이용하여 부호화 하였을 때 최종 움직임 벡터와 각 예측 벡터 간의 거리를 나타낸 것이다. 영벡터는 표에서 빠져 있는데 이는 다음 절에서 설명하도록 한다.

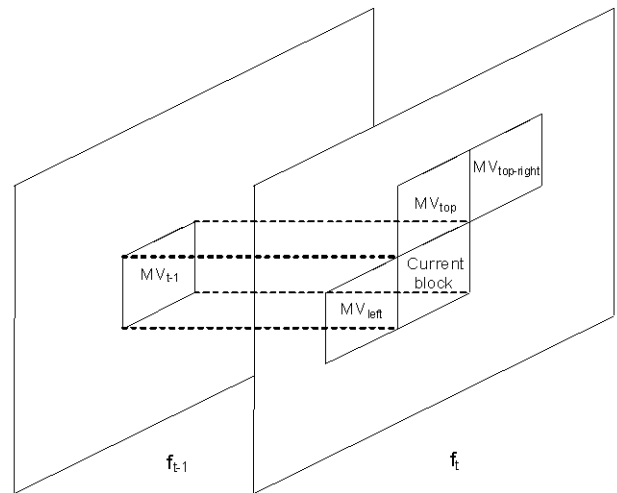


그림 3. 제안하는 알고리즘에서 탐색 시작점을 결정하기 위해 사용하는 4개의 예측 벡터

Fig. 3. 4 predictors used to predict the starting point in proposed algorithm

표 1. PMVFAST를 사용하여 전체 프레임을 부호화하였을 때 최종 움직임 벡터와 각 예측 벡터 간의 평균거리.

Table 1. Distance in pixels from each predictor to the final motion vector in PMVFAST

| Sequence     | MVleft  | MVtop   | MVtop-right | MVt-1   |
|--------------|---------|---------|-------------|---------|
| Bus          | 1.44806 | 1.64125 | 1.88135     | 0.81642 |
| Coastguard   | 0.73289 | 0.63438 | 0.73778     | 0.57917 |
| Foreman      | 1.39627 | 1.29688 | 1.63483     | 1.53961 |
| Stefan       | 2.25219 | 2.50256 | 2.79508     | 2.08515 |
| Table Tennis | 0.72754 | 0.71724 | 0.88137     | 0.57195 |

표 1은 Foreman을 제외한 전 시퀀스에서  $MV_{t-1}$ 이 최종 움직임 벡터와 가장 유사함을 보여준다. 이는  $MV_{t-1}$ 을 시작점 예측에 사용하면 보다 정확한 시작점 예측을 할 수 있다는 것을 뜻한다. 따라서 제안하는 알고리즘은 탐색 시작점 ( $d_x, d_y$ )를 다음과 같은 식으로 구한다.

$$\begin{cases} d_x = \frac{1}{2} \left\{ \sum_i MV_x^i - \max(MV_x^i) - \min(MV_x^i) \right\} \\ d_y = \frac{1}{2} \left\{ \sum_i MV_y^i - \max(MV_y^i) - \min(MV_y^i) \right\} \end{cases}, (2)$$

$i \in \{left, top, top-right, t-1\}$

즉, 4개의 예측 벡터의 x, y 중 가장 큰 값과 가장 작은 값을 제외한 나머지 2개의 예측 벡터의 평균값으로서 시작점의 좌표  $d_x, d_y$ 가 독립적으로 결정된다.

프레임의 첫 번째 열에 존재하는 매크로블록의 경우,  $MV_{left}$ 가 존재하지 않으므로 사용할 수 있는 움직임 벡터가 3개이다. 이 경우에는 다음과 같이 기존의 중간값 예측을 그대로 사용한다.

$$\begin{cases} d_x = median(MV_x^{top}, MV_x^{top-right}, MV_x^{t-1}) \\ d_y = median(MV_y^{top}, MV_y^{top-right}, MV_y^{t-1}) \end{cases} (3)$$

프레임의 첫 번째 행에 존재하는 매크로블록의 경우  $MV_{top}$ 과  $MV_{top-right}$ 가 존재하지 않는다. 이 경우에는 이용할 수 있는 예측 벡터가 2가지 밖에 없기 때문에 영벡터를 추가하여 기존의 중간값 예측을 사용한다.

$$\begin{cases} d_x = median(MV_x^{left}, MV_x^{t-1}, 0) \\ d_y = median(MV_y^{left}, MV_y^{t-1}, 0) \end{cases} (4)$$

프레임의 첫 번째 매크로블록, 즉 (0, 0)에 위치한 매크로블록의 경우에는 존재하는 예측 벡터가  $MV_{t-1}$  뿐이므로 이것을 그대로 사용한다.

탐색 시작점에서의 SAD가 충분히 작다면, 즉 문턱값  $T_1$ 보다 작다면 더 이상 탐색을 하지 않고 탐색 시작점이 최종 움직임 벡터가 된다. 실험적으로  $T_1$ 은 256으로 결정하였다. 게다가 표 1의 결과에서 볼 수 있듯이  $MV_{t-1}$ 과 최종 움직임 벡터는 상당히 유사하므로, 탐색 시작점이  $MV_{t-1}$ 과 같다면

더 큰 문턱값을 사용할 수도 있다. 따라서 이 경우에는 다음과 같이  $T_1$ 을 결정한다.

$$T_1 = \max(SAD_{t-1}, 256) \quad (5)$$

표 2는 PMVFAST와 제안한 알고리즘을 사용하였을 때, 탐색 시작점이 최종 움직임 벡터가 되는 횟수를 비교한 것이다. 제안한 알고리즘이 PMVFAST에 비해 훨씬 높은 수치를 보여주는데, 두 알고리즘의 종료조건이 같으므로 제안하는 시작점 예측이 PMVFAST의 시작점 예측보다 더 정확함을 알 수 있다.

표 2. PMVFAST와 제안하는 알고리즘을 이용하여 전체 프레임을 부호화하였을 때, 탐색 시작점이 최종 움직임 벡터로 선택되는 횟수  
Table 2. Frequency that the search is ended after just one point is checked.

| Sequence     | PMVFAST | Proposed |
|--------------|---------|----------|
| Bus          | 13542   | 15391    |
| Coastguard   | 27941   | 32403    |
| Foreman      | 24660   | 27205    |
| Stefan       | 23572   | 24513    |
| Table Tennis | 46101   | 47066    |

## 2. 예측벡터로서 영벡터의 효율성

PMVFAST는 시작점에서 탐색이 종료되지 않을 경우 5개의 예측 벡터를 더 사용하는데,  $MV_{left}, MV_{top}, MV_{top-right}, MV_{t-1}$ , 영벡터가 그것이다. 표 3은 영벡터를 제외한 나머지 4개의 예측 벡터가 모두 영벡터가 아닐 때 최종적으로 영벡터가 선택되는 횟수와 확률을 구한 것이다.

표 3을 보면 예측 벡터들이 영벡터가 아닐 때 영벡터가 선택될 확률은 극히 낮음을 알 수 있다. 이는 현재 매크로블록의 움직임 벡터가 영벡터라면, 대부분의 경우 예측 벡터 중 적어도 하나가 영벡터임을 의미한다. 즉, 대부분의 경우 영벡터는 검사할 필요가 없는데도 검사하고 있으며, 표 4를 보면 영벡터를 검사하는 것이 속도와 화질을 오히려 저하시킨다는 것을 알 수 있다. 여기에서 Speed-up은 전역 탐색 기법을 사용했을 때의 탐색점 수와 각 알고리즘을 사용했

표 3. MVleft, MVtop, MVtop-right, MVt1이 영벡터가 아닐 때, 최종적으로 영벡터가 선택된 횟수와 확률

Table 3. Frequency and probability that the zero predictor is selected what any predictor is not the zero vector.

|              | 횟수   | 확률 (%) |
|--------------|------|--------|
| Bus          | 1282 | 0.023  |
| Coastguard   | 3472 | 0.031  |
| Foreman      | 2828 | 0.026  |
| Stefan       | 4540 | 0.041  |
| Table Tennis | 1178 | 0.011  |

을 때의 탐색점 수의 비율로써, 숫자가 클수록 빠르다고 할 수 있다.

영벡터를 포함하였을 때는 좀 더 낮은 SAD를 갖는 블록을 찾아낼 수 있지만, 영벡터를 제외하였을 때에는 움직임 벡터 간의 상관성이 더 높아진다. 표 4의 실험결과는 더 낮은 SAD를 갖는 블록을 찾는 것보다 움직임 벡터 간의 상관성을 높이는 것이 좀 더 유리함을 보여 주고 있다.

따라서 제안하는 알고리즘은 첫 번째 종료조건을 만족하지 않았을 경우 영벡터를 제외한 4개의 예측 벡터, 즉 MVleft, MVtop, MVtop-right, MVt1을 더 검사한다. 4개의 예측 벡터를 탐색한 후 최소의 SAD를 갖는 예측 벡터가 다이아몬드 탐색 시작점이 된다. 다이아몬드 탐색 시작점이 결정되면 문턱값 T2가 적용된다. 만약 다이아몬드 탐색 시작점에서의 SAD가 T2보다 작다면 더 이상 탐색을 하지 않고 종료되는데, T2는 다음과 같이 계산된다.

$$T_2 = \min(SAD_{left}, SAD_{top}, SAD_{top-right}) \quad (6)$$

T2가 너무 작은 값이나 너무 큰 값으로 선택되지 않게 하기 위하여 T2를 512와 1024 사이의 값으로 제한한다.

탐색 시작점을 결정할 때와 마찬가지로 최종 움직임 벡터

와 MVt1과의 상관성이 매우 크기 때문에 다이아몬드 탐색 시작점과 MVt1이 같다면 T2를 다음과 같이 조정한다.

$$T_2' = \max(SAD_{i-1}, T_2) \quad (7)$$

만약 SAD가 이 문턱값보다 크다면 SDSP를 사용하여 탐색을 계속 진행한다. 탐색점의 수를 줄이기 위하여 LDSP는 사용하지 않는다.

#### IV. 실험결과

실험은 MPEG-4 VM 인코더[12]를 이용하여 이루어졌다. 모든 영상은 IPPP로 부호화 되었고, 비트율 제어(rate control)는 TM5를 적용하였다. 실험에 사용한 영상은 QCIF (176×144), CIF (352×288), SIF (320×240)의 3종류이며, 각 영상에서의 비트율과 프레임율, 탐색범위는 [13]에 따라 다양한 환경에서 실험을 하였다.

움직임 추정은 화질과 부호화 속도에 영향을 주기 때문에 알고리즘의 성능은 PSNR과 속도의 측면에서 측정하였다. 여기에서 PSNR은 전 프레임에서의 평균 PSNR이고, 속도향상비는 전역 탐색 기법을 사용했을 때와 각 알고리즘을 사용하였을 때의 탐색점의 수의 비율이다. 지면이 한정되어 있고, MVFAST와 PMVFAST의 성능이 다른 알고리즘에 비해 월등히 높으므로 다른 알고리즘과의 비교는 생략하고 이 2가지의 알고리즘과의 비교만을 제시한다.

표 5의 실험결과를 보면, 제안한 알고리즘은 모든 경우에서 MVFAST와 PMVFAST보다 빠른 것을 알 수 있다. 제안한 알고리즘은 MVFAST보다 평균적으로 약 60%가 빠르고, PMVFAST보다는 약 10%가 빠르다. 게다가 평균

표 4. 영벡터를 포함시켰을 때와 제외시켰을 때 각 시퀀스에서의 평균 PSNR과 전역탐색기법 대비 속도향상비율.  
Table 4. Average PSNR and speed-up ratio with and without zero predictor.

|           | FS    | MVFAST | PMVFAST | Proposed (영벡터포함) | Proposed (영벡터제외) |
|-----------|-------|--------|---------|------------------|------------------|
| PSNR (dB) | 31.11 | 31.10  | 31.11   | 31.14            | 31.16            |
| Speed-up  | 1     | 410.37 | 600.46  | 631.58           | 652.42           |

표 5. 전역 탐색 기법, MVFAST, PMVFAST과 비교한 실험 결과

Table 5. Simulation results of proposed algorithm compared with FS, MVFAST, and PMVFAST.

| Sequence          | Format | Bit-rate (kbps) | Frame Rate | Search Range | Measure   | Motion Estimation Algorithms |         |         |          |
|-------------------|--------|-----------------|------------|--------------|-----------|------------------------------|---------|---------|----------|
|                   |        |                 |            |              |           | FS                           | MVFAST  | PMVFAST | Proposed |
| Container         | QCIF   | 10              | 7.5        | 16           | PSNR (dB) | 27.67                        | 27.98   | 27.96   | 27.86    |
|                   |        |                 |            |              | Speed-up  | 1                            | 218.96  | 285.33  | 305.59   |
|                   |        |                 |            | 32           | PSNR (dB) | 27.89                        | 27.90   | 27.85   | 27.90    |
|                   |        |                 |            |              | Speed-up  | 1                            | 787.58  | 1053.63 | 1087.18  |
| Hall Monitor      | QCIF   | 10              | 7.5        | 16           | PSNR (dB) | 27.36                        | 27.37   | 27.65   | 27.64    |
|                   |        |                 |            |              | Speed-up  | 1                            | 211.93  | 317.22  | 357.09   |
|                   |        |                 |            | 32           | PSNR (dB) | 27.28                        | 27.35   | 27.62   | 27.66    |
|                   |        |                 |            |              | Speed-up  | 1                            | 763.74  | 1125.91 | 1294.16  |
| Mother & Daughter | QCIF   | 24              | 10         | 16           | PSNR (dB) | 33.75                        | 33.54   | 33.64   | 33.66    |
|                   |        |                 |            |              | Speed-up  | 1                            | 208.21  | 298.11  | 310.14   |
|                   |        |                 |            | 32           | PSNR (dB) | 33.65                        | 33.60   | 33.70   | 33.70    |
|                   |        |                 |            |              | Speed-up  | 1                            | 757.16  | 1086.82 | 1117.33  |
| Silent            | QCIF   | 24              | 10         | 16           | PSNR (dB) | 28.79                        | 28.88   | 28.94   | 28.98    |
|                   |        |                 |            |              | Speed-up  | 1                            | 137.28  | 184.26  | 209.79   |
|                   |        |                 |            | 32           | PSNR (dB) | 28.77                        | 28.97   | 28.98   | 28.92    |
|                   |        |                 |            |              | Speed-up  | 1                            | 493.09  | 662.18  | 744.13   |
| Coastguard        | QCIF   | 48              | 10         | 16           | PSNR (dB) | 28.52                        | 28.51   | 28.51   | 28.53    |
|                   |        |                 |            |              | Speed-up  | 1                            | 101.82  | 158.41  | 182.54   |
|                   |        |                 |            | 32           | PSNR (dB) | 28.53                        | 28.53   | 28.52   | 28.55    |
|                   |        |                 |            |              | Speed-up  | 1                            | 367.26  | 564.70  | 651.32   |
| Foreman           | CIF    | 112             | 10         | 16           | PSNR (dB) | 29.32                        | 29.27   | 29.26   | 29.26    |
|                   |        |                 |            |              | Speed-up  | 1                            | 84.52   | 103.33  | 105.71   |
|                   |        |                 |            | 32           | PSNR (dB) | 29.53                        | 29.43   | 29.45   | 29.46    |
|                   |        |                 |            |              | Speed-up  | 1                            | 306.24  | 375.59  | 387.78   |
| Coastguard        | CIF    | 112             | 10         | 16           | PSNR (dB) | 26.94                        | 26.98   | 26.97   | 26.99    |
|                   |        |                 |            |              | Speed-up  | 1                            | 98.20   | 111.25  | 121.82   |
|                   |        |                 |            | 32           | PSNR (dB) | 26.96                        | 27.00   | 27.03   | 27.03    |
|                   |        |                 |            |              | Speed-up  | 1                            | 373.58  | 426.15  | 464.30   |
| Foreman           | CIF    | 512             | 15         | 16           | PSNR (dB) | 34.99                        | 34.95   | 35.00   | 35.00    |
|                   |        |                 |            |              | Speed-up  | 1                            | 98.51   | 139.87  | 148.85   |
|                   |        |                 |            | 32           | PSNR (dB) | 35.17                        | 35.11   | 35.17   | 35.18    |
|                   |        |                 |            |              | Speed-up  | 1                            | 368.02  | 527.49  | 563.45   |
|                   |        |                 |            | 48           | PSNR (dB) | 35.18                        | 35.12   | 35.16   | 35.18    |
|                   |        |                 |            |              | Speed-up  | 1                            | 789.30  | 1122.85 | 1202.96  |
| Foreman           | CIF    | 1024            | 30         | 16           | PSNR (dB) | 35.87                        | 35.84   | 35.88   | 35.88    |
|                   |        |                 |            |              | Speed-up  | 1                            | 127.71  | 205.57  | 222.49   |
|                   |        |                 |            | 32           | PSNR (dB) | 38.09                        | 38.03   | 38.08   | 38.09    |
|                   |        |                 |            |              | Speed-up  | 1                            | 369.15  | 548.00  | 585.42   |
|                   |        |                 |            | 48           | PSNR (dB) | 35.88                        | 35.85   | 35.89   | 35.89    |
|                   |        |                 |            |              | Speed-up  | 1                            | 1028.73 | 1646.50 | 1779.69  |
| Table Tennis      | SIF    | 1024            | 30         | 16           | PSNR (dB) | 31.77                        | 31.73   | 31.75   | 31.75    |
|                   |        |                 |            |              | Speed-up  | 1                            | 137.36  | 226.61  | 251.86   |
|                   |        |                 |            | 32           | PSNR (dB) | 31.76                        | 31.71   | 31.74   | 31.75    |
|                   |        |                 |            |              | Speed-up  | 1                            | 516.55  | 848.29  | 934.35   |
|                   |        |                 |            | 48           | PSNR (dB) | 31.77                        | 31.69   | 31.73   | 31.74    |
|                   |        |                 |            |              | Speed-up  | 1                            | 1093.70 | 1792.42 | 1977.80  |

PSNR도 전역 탐색 기법보다 높은 것을 볼 수 있다. 특히 저비트율에서의 성능이 좋은데, 이는 저비트율에서는 움직임

임 벡터의 상관성이 중요하고, 고비트율에서는 낮은 SAD를 갖는 블록을 찾는 것이 중요하다는 것을 뜻한다. 하지만 제안하는 알고리즘은 정확한 탐색 시작점 예측을 바탕으로 초기 종료 조건을 줄였기 때문에 고비트율에서도 전역 탐색 기법에 거의 근접한 PSNR을 보여주고 있다.

### V. 결론

이 논문에서는 기존의 알고리즘보다 정확하게 탐색 시작점을 예측함으로써 효율적인 부호화를 가능하게 하는 고속 움직임 추정 기법을 제안하였다. 제안된 알고리즘은 예측 움직임 벡터의 시간적, 공간적 상관성을 이용하여 탐색 시작점을 예측하기 때문에 예측이 매우 정확하고, 따라서 속도와 화질 모두를 개선할 수 있다.

정량적인 측면에서 MVFAST의 경우 추가적인 메모리가 필요하지 않지만 전역 탐색 기법에 비해 낮은 PSNR을 보이고, PMVFAST는 이전 프레임에서의 움직임 벡터를 저장하기 위한 메모리가 필요하지만 전역 탐색 기법과 거의 유사한 PSNR과 높은 속도 개선 효과를 보인다. 제안하는 알고리즘은 PMVFAST와 같은 양의 메모리를 사용하면서도 더 빠르고 전역 탐색 기법보다도 높은 평균 PSNR을 보여준다. 즉, 본 알고리즘은 PMVFAST와 같은 복잡도를 가지지만 화질과 속도 면에서 더 우수한 알고리즘이라고 할 수 있다.

### 참 고 문 헌

[1] M. Ghanbari, Standard Codec: Image Compression to Advanced

Video Coding, IEE, 2003.  
 [2] I. Richardson, H.264 and MPEG-4 Video Compression, Wiley, 2003.  
 [3] G. Sorwar, M. Mushed, and L. Dooley, "Block-based true motion estimation using distance dependent thresholded search," Proc. ISCA Computer Applications in Industry and Engineering, pp. 45-48, 2001.  
 [4] J. R. Jain, and A. K. Jain, "Displacement measurement and its application in interframe image coding," IEEE Trans. On Communications, Vol. 12, pp. 1799-1808, Dec. 1981.  
 [5] T. Koga, K. Inuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," Proc. Nat. Telecommun. Conf., New Orleans, LA., G5.3.1-G5.3.5, Dec 1981.  
 [6] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," IEEE Trans. Circuits and Systems for Video Technology, Vol. 4, pp. 438-442, Aug. 1994.  
 [7] S. Zhu, and K. K. Ma, "A new diamond search algorithm for fast block matching motion estimation," IEEE Trans. Image Processing, Vol. 9, No. 2, pp. 287-290, Feb. 2000.  
 [8] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," IEEE Trans. Circuits and System for Video Technology, Vol. 8, No. 4, pp. 369-377, Aug. 1998.  
 [9] P. I. Hosur and K. K. Ma, "Motion vector field adaptive fast motion estimation," Proc. Int. Conf. Information Communications and Signal Processing (ICICSP), Singapore, Dec. 1999.  
 [10] A. M. Tourapis, O. C. Au, and M. L. Liou, "Predictive motion vector field adaptive search technique (PMVFAST) - Enhancing block based motion estimation," Proc. Visual Comm. and Image Proc. (VCIP), San Jose, CA., Jan. 2001.  
 [11] Text of 14496-7 PDTR (Optimized visual reference software), ISO/IEC N4057, 2001.  
 [12] MPEG Video Group, "The MPEG-4 video verification model version 18.0," ISO/IEC JTC1/SC29/WG11 N3908, 2001.  
 [13] Implementation Study Group, "Experimental conditions for evaluating encoder motion estimation algorithms," ISO/IEC JTC1/SC29/WG11 N3141, 1999.

### 저 자 소 개



최 민 석

- 2006년 2월 : 한양대학교 전자통신컴퓨터공학과 졸업
- 2006년 3월~현재 : 한양대학교 전자통신컴퓨터공학과 석사과정
- 주 관심분야 : 영상압축 및 후처리

---

 저 자 소 개
 

---



김 종 호

- 1998년 2월 : 한양대학교 전자통신공학과 졸업
- 2000년 2월 : 한양대학교 전자통신공학과 석사
- 2000년 3월~현재 : 한양대학교 전자통신컴퓨터공학과 박사과정
- 주관심분야 : 영상처리 및 영상압축, 후처리



정 제 창

- 1980년 2월 : 서울대학교 전기공학과 졸업
- 1982년 2월 : KAIST 전자전기공학과 석사
- 1990년 : 미국 미시간 대학 전기공학과 공학박사
- 1980년~1986년 : KBS 기술연구소 연구원 (디지털 TV 및 뉴미디어 연구)
- 1990년~1991년 : 미국 미시간 대학 전기공학과 연구교수 (영상 및 신호처리 연구)
- 1991년~1995년 : 삼성전자 멀티미디어 연구소 (MPEG, HDTV, 멀티미디어 연구)
- 1995년~현재 : 한양대학교 전자통신컴퓨터공학과 교수 (영상통신 및 신호처리 연구실)
- 1998년 11월 27일 : 과학기술자상 수상
- 1998년 12월 31일 : 정보통신부장관상 표창
- 주관심분야 : 영상처리 및 영상압축