

# 동영상 압축을 위한 고속 반화소 단위 블록 정합 알고리즘

정희원 이범기\*, 정원식\*, 김덕규\*

## Fast Block Matching Algorithm With Half-pel Accuracy for Video Compression

Bub-Ki Lee\*, Won-Sik Cheong\*, Duk-Gyoo Kim\* *Regular Members*

### 요 약

본 논문에서는 반화소 위치에서의 평균 절대 오차 (mean absolute difference; MAD)의 최소 범위를 이용한 고속 반화소 단위의 블록 정합 알고리즘을 제안한다. 제안한 방법에서는 반화소 단위의 MAD의 최소 범위를 화소 단위의 움직임 추정으로 얻어진 MAD들을 이용하여 구한 후, 반화소 단위의 MAD의 최소 범위가 화소 단위의 움직임 추정으로 구한 기준 MAD보다 작은 탐색점에 대해서만 블록 정합을 행하였다. 제안한 방법의 성능을 평가하기 위한 컴퓨터 모의 실험결과로부터 제안한 방법이 기존의 방법과 동일한 성능을 유지하면서도 많은 계산량을 줄일 수 있음을 확인할 수 있었다.

### ABSTRACT

In this paper, we propose the fast block matching algorithm with half pel accuracy using the lower bound of mean absolute difference (MAD) at search point of half pel accuracy motion estimation. The proposed method uses the lower bound of MAD at search point of half pel accuracy which calculated from MAD's at search points of integer pel accuracy. We can reduce the computational complexity by executing the block matching operation only at the necessary search points. The points are selected when the lower bound of MAD at that point is smaller than reference MAD of integer pel motion estimation. Experimental results show that the proposed method can reduce the computational complexity considerably and keeping the same performance with conventional method.

### I. 서론

동영상 압축 (video compression)은 중요한 디지털 기술의 하나로서 고선명 TV (high definition TV; HDTV), 주문형 비디오 (video on demand), 영상 회의 (video conferencing) 등의 다양한 응용 분야에 사용된다. 일반적으로 동영상 압축 방법은 동일 프레임내의 화소들 사이에 존재하는 공간적

중복성을 제거하는 부분, 인접 프레임간의 시간적 중복성을 제거하는 부분 및 발생 부호의 중복성을 제거하는 부분으로 구성된다. 여기서 시간적인 중복성을 줄이기 위하여 사용되는 움직임 추정 및 보상은 동영상 압축에 있어서 높은 압축율을 얻는데 핵심적인 역할을 담당하고 있으며, 동영상 압축을 위한 표준인 H.261,<sup>[1]</sup> H.263,<sup>[2]</sup> MPEG<sup>[3],[4]</sup> 등에 사용되고 있다.

움직임 추정 기법은 크게 화소 순환 알고리즘

\* 경북대학교 전자전기공학부(khlee@palgong.kyungpook.ac.kr)  
논문번호 : 99115-0328, 접수일자 : 1999년 3월 28일

(pel recursive algorithm; PRA)과 블록 정합 알고리즘 (block matching algorithm; BMA)으로 나누어진다. 이때 BMA는 PRA에 비하여 수행시간이 적게 소요되고 하드웨어 구현이 용이하기 때문에 기존의 압축 표준에서 많이 사용되고 있다.<sup>[5]-[7]</sup> 이 방법에서는 입력 영상을 임의의 작은 블록으로 나눈 뒤, 블록내의 모든 화소들은 같은 방향으로 평행 이동한다는 것을 가정하여 이전 프레임의 탐색영역에서 정합 척도가 최적인 블록을 찾는다. BMA를 이용하여 움직임을 추정하는 경우에 가장 좋은 성능을 얻을 수 있는 방법은 이전 프레임의 탐색영역의 모든 탐색점에 대하여 탐색을 행하는 전역 탐색 블록 정합 알고리즘 (full search BMA; FSBMA)이다. FSBMA는 모든 탐색점에 대하여 탐색을 행하므로 계산량이 많은 단점이 있지만, 정합 척도가 최적인 블록을 찾기 때문에 움직임 추정 오차 측면에서 최적인 움직임 벡터를 얻을 수 있으므로 MPEG-2등에서 권고하고 있다.

그러나 실제 동영상에서 물체의 움직임은 반드시 샘플링 격자 간격 (sampling grid distance)의 정수 배로만 일어나는 것이 아니므로 더욱 정확한 움직임 추정 및 보상을 위해 분수 화소 (fractional pel) 단위의 움직임 추정 및 보상 방법이 사용되고 있다.[8]-[13] Girod[9]는 TV신호와 비디오 폰 (video phone)신호에서 반화소, 1/4화소, 그리고 1/8화소 단위의 움직임 추정에 대한 성능을 평가하였다. 이 결과로부터, TV 신호 및 비디오 폰 신호는 각각 1/4 화소 단위 및 반화소 단위의 움직임 추정으로 충분한 성능을 얻을 수 있음을 알 수 있다.  $I_u^{[10]}$ 는 공간 영역에서 화소 단위로 움직임 추정을 행한 후, 이로부터 얻어진 화소 단위 움직임 벡터를 기준으로 탐색 영역의 크기를 축소시켜 반화소 단위의 움직임 추정을 행하였다. 그러나 이 방법은 정확한 움직임을 찾을 수 있지만 탐색점 수와 보간량의 증가로 계산량이 매우 많다. MPEG-2<sup>[11]</sup>에서는 FSBMA로 최적의 화소 단위의 움직임 벡터를 찾은 후, 화소 단위의 움직임 벡터를 기준점으로 하여 이전 프레임의 영상을 보간하여 만든 이웃 8개의 반화소 위치와 기준점을 포함하여 정합 척도가 최적인 위치를 찾는다. 이 방법에서는 이전 프레임내의 화소와 화소 사이를 보간하여 더욱 정밀하게 움직임을 추정하지만 보간 및 부가적인 블록 정합에 의한 계산량이 증가하는 단점이 있다.

이러한 계산량의 증가를 해결하기 위하여, 화소 단위로 찾은 움직임 벡터 주위의 움직임 추정 오차

들을 이용한 모델로부터 반화소 단위의 움직임임을 추정하는 방법이 제안되었다.<sup>[12][13]</sup> 이 방법들에서는 보간이 필요 없으므로 매우 많은 계산량을 줄일 수 있었다. 그러나 이 방법들은 블록 정합을 행하지 않기 때문에 움직임 추정 오차가 커지는 단점이 있으며, 특히 영상의 특성이 모델과 맞지 않는 경우에는 반화소 단위의 움직임 추정 오차가 화소 단위의 움직임 추정 오차보다 더 커지게 되는 단점이 있다. 그러므로 MPEG-2의 알고리즘에 동일한 성능을 유지하면서 계산량을 줄일 수 있는 반화소 단위의 움직임 추정 방법이 필요하다.

본 논문에서는 반화소 위치에서의 MAD의 최소 범위를 이용한 고속 반화소 단위의 블록 정합 알고리즘을 제안한다. 제안한 방법에서는 반화소 단위의 MAD의 최소 범위를 화소 단위의 움직임 추정으로 얻어진 MAD들을 이용하여 구한 후, 반화소 단위의 MAD의 최소 범위가 화소 단위의 움직임 추정으로 구한 기준 MAD보다 작은 탐색점에 대해서만 블록 정합을 행하였다. 즉 화소 단위의 움직임 벡터의 주위 8개 탐색점에 대하여 MAD의 최소 범위를 구한 뒤, 블록 정합이 필요한 경우에만 정합을 행하므로써 MPEG-2의 방법과 동일한 성능을 유지하면서 계산량을 줄였다.

제안한 방법의 성능을 평가하기 위한 컴퓨터 모의 실험결과로부터 제안한 방법이 기존의 방법과 동일한 성능을 유지하면서도 많은 계산량을 줄일 수 있음을 확인할 수 있었다.

## II. 반화소 단위 움직임 추정 기법

인접 프레임간의 시간적인 중복성을 줄이기 위한 움직임 추정 및 보상 부호화기에서는 더욱 정확한 움직임 추정을 통하여 높은 압축 효율을 얻기 위해서 분수 단위의 화소 단위 움직임 추정 기법이 행해지고 있다. 이는 실제 동영상에서 움직이는 물체는 반드시 카메라의 샘플링 격자 간격의 정수 배로만 움직이지 않는다는 사실에 근거한다. 즉 현재 프레임과 이전 프레임과 관계를

$$I_t(i, j) = I_{t-1}(i - d_i, j - d_j) \quad (1)$$

와 같이 나타낼 때, 수평 및 수직 방향으로의 실제의 움직임 거리 (displacement distance)  $d_i$  및  $d_j$ 는

$$d_i = (m_i + v_i)a \quad (2)$$

$$d_j = (m_j + v_j)a$$

와 같이 반드시 샘플링 간격  $a$ 의 정수 배로만 나타나지 않는다. 여기서  $I_t(i, j)$ 는 현재 프레임의  $(i, j)$ 위치에서 화소의 휘도 값,  $I_{t-1}(i-d_i, j-d_j)$ 는 이전 프레임의  $(i-d_i, j-d_j)$ 에서 화소의 휘도 값이다. 그리고  $m_i$ 와  $m_j$ 는 움직임 거리의 정수 성분이고,  $v_i$  및  $v_j$ 는 소수 부분을 나타낸 것으로  $-0.5$ 에서  $0.5$ 사이의 값을 갖는다.

MPEG-2의 부호화기에서는 화소 단위로만 탐색할 때 나타내는 움직임 오차를 더욱 줄임으로서 높은 압축률을 얻기 위하여 반화소 단위의 움직임 추정을 행한다. MPEG-2의 부호화기에서 사용된 반화소 단위의 움직임 추정 기법에서는 FSBMA로 최적의 화소 단위의 움직임 벡터를 찾은 후, 화소 단위 움직임 벡터를 기준점으로하여 그림 1에서와 같이 이전 프레임의 영상을 보간(interpolation)하여 이웃 8개의 반화소 위치를 만든 뒤, 기준점과 이웃의 8개의 반화소 위치에 대하여 정합 척도가 최적인 위치를 찾는다. 이때 사용되는 정합 척도로는 평균 제곱 오차(mean squared difference; MSD)와 비슷한 성능을 유지하면서도 계산량이 적고 하드웨어 구현이 용이한 MAD가 널리 이용되고 있다. 이때, MAD는

$$MAD_{(k,l)}(v_x, v_y) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} | I_t(k+i, l+j) - I_{t-1}(k+v_x+i, l+v_y+j) | \quad (3)$$

와 같다. 여기서,  $N$ 은 블록의 수직 및 수평 방향의 크기,  $x$  및  $y$ 는 FSBMA로 추정된 화소 단위 움직임 벡터,  $(k, l)$ 은 블록의 위치 좌표, 그리고  $(v_x, v_y)$ 는 반화소 단위의 탐색점의 위치를 나타낸다. 그리고  $v_x$  및  $v_y$ 는 각각  $-0.5, 0$ , 그리고  $0.5$ 의 값을 가진다. 이를 이용하여  $(k, l)$ 번째 블록의 반화소 단위 움직임 벡터는

$$MV^h(k, l) = \arg \min_{(v_x, v_y)} MAD_{(k,l)}(v_x, v_y) \quad (4)$$

와 같이 구한다. 또한 보간을 위한 기법으로는 양선형(bilinear), Lagrange 등 여러 가지 기법이 있으나, 이들 중 양선형 기법을 많이 사용하고 있으며, 이를 수식으로는

$$I_{t-1}(x, y \pm 0.5) = [I_{t-1}(x, y) + I_{t-1}(x, y \pm 1)]/2$$

$$I_{t-1}(x \pm 0.5, y \pm 0.5) = [I_{t-1}(x, y) + I_{t-1}(x \pm 1, y) + I_{t-1}(x, y \pm 1) + I_{t-1}(x \pm 1, y \pm 1)]/4 \quad (5)$$

$$I_{t-1}(x \pm 0.5, y) = [I_{t-1}(x \pm 1, y) + I_{t-1}(x, y)]/2$$

와 같이 표현 할 수 있다.

블록의 크기가  $N \times N$ 일 때, 반화소 위치에서의 MAD는 이전 프레임에서의 보간 블록과 현재 프레임 블록의 차의 합으로 구할 수 있으므로, 블록 당  $2N^2$  번의 덧셈 계산이 필요하다. 즉, 한 블록에 대한 반화소 단위의 탐색점 수는 8개이고, 한 프레임의 블록의 개수가  $T$  이면, 한 프레임에 대한 반화소 단위의 움직임 추정을 위한 계산량은  $2 \times 8 \times T \times N^2$ 이고, 한 프레임을 보간하는데 필요한 계산량은  $5TN^2$ 이므로 총계산량은  $16TN^2 + 5TN^2$ 이지만 보간을 위한 계산은 알고리즘에 관계없이 동일하게 적용되므로 반화소 단위의 움직임 추정에 필요한 계산량은

$$C_{Halfpel} = 16TN^2 \quad (6)$$

와 같다. 여기서, 덧셈과 뺄셈은 동일 계산량으로 취급하고, 2의 제곱형태는 이동 연산(shift operation)으로 쉽게 구현이 가능하므로 계산량에서 제외하였다. 따라서 반화소 단위의 움직임 추정 방법은 정확한 움직임 추정을 통하여 예측오차 분산

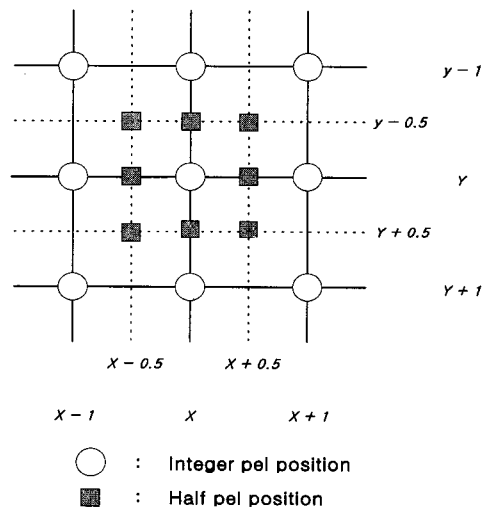


그림 1. 반화소 변위

이 줄어들어 압축 효율을 높일 수 있으나, FSBMA에 필요한 계산량 이외에 별도의 계산량이 필요하다는 단점을 있다. 이러한 계산량을 줄이기 위한 방법으로 화소 단위로 찾은 움직임 벡터 주위의 MAD들을 이용한 모델로부터 반화소 단위의 움직임 벡터를 추정하는 방법이 제안하였다.<sup>[12][13]</sup> 이 방법들은 화소간의 보간 및 블록 정합을 행하지 않고 반화소 움직임 벡터를 찾기 때문에 MPEG-2의 알고리즘에 비해 많은 계산량을 줄일 수 있었다. 그러나 이 방법들은 블록 정합을 행하지 않기 때문에 움직임 추정 오차가 커지는 단점이 있으며, 특히 영상의 특성이 모델과 맞지 않는 경우에는 반화소 단위의 움직임 추정 오차가 화소 단위의 움직임 추정 오차보다 더 커지게 되는 단점이 있다. 그러므로 MPEG-2의 알고리즘에 동일한 성능을 유지하면서 계산량을 줄일 수 있는 반화소 움직임 추정 방법이 필요하다.

### III. 제안한 고속 반화소 단위 블록 정합 알고리즘

본 논문에서는 반화소 단위의 MAD의 최소 범위를 화소 단위의 움직임 추정에서 얻어진 MAD들을 이용하여 구한 후, 이를 이용하여 탐색점 수를 줄임으로써 고속으로 반화소 단위의 움직임 추정을 행할 수 있는 기법을 제안한다.

현재 프레임  $(k, l)$  위치에서  $N \times N$  블록을 블록 A라할때, 블록 A를  $N \times N$  행렬로 나타내면

$$A = \begin{bmatrix} I_A(k, l) & \dots & I_A(k, l+N-1) \\ \vdots & & \vdots \\ I_A(k+N-1, l) & \dots & I_A(k+N-1, l+N-1) \end{bmatrix} \quad (7)$$

와 같다. 또한 그림 1에 나타낸 바와 같이 블록 A의 화소 단위의 움직임 벡터 위치  $(x, y)$ 의 블록, 즉 이전 프레임에서  $(k+x, l+y)$  위치의 블록  $S(x+0.5, y)$ 를  $N \times N$  행렬로 나타내면

$$S(x, y) = \begin{bmatrix} I_{t-1}(k+x, l+y) & \dots & I_{t-1}(k+x, l+y+N-1) \\ \vdots & & \vdots \\ I_{t-1}(k+x+N-1, l+y) & \dots & I_{t-1}(k+x+N-1, l+y+N-1) \end{bmatrix} \quad (8)$$

와 같다. 이때, 반화소 단위의 움직임 탐색점

$(x+0.5, y)$ 에서 블록 A의 정합에 의한 오차 블록  $E(x+0.5, y)$ 는

$$E(x+0.5, y) = A - S(x+0.5, y) \quad (9)$$

와 같이 표현 할 수 있다. 여기서,  $S(x+0.5, y)$ 는 식 (5)로부터  $\frac{1}{2}[S(x, y) + S(x+1, y)]$  이므로, 이를 식 (9)에 대입하면

$$E(x+0.5, y) = A - \frac{1}{2}[S(x, y) + S(x+1, y)] \quad (10)$$

와 같다. 또한, 블록 A와 블록  $S(x, y)$  및 블록 A와 블록  $S(x+1, y)$ 의 정합에 의한 오차 블록  $E(x, y)$  및  $E(x+1, y)$ 는

$$E(x, y) = A - S(x, y) \quad (11)$$

$$E(x+1, y) = A - S(x+1, y) \quad (12)$$

이다. 이때, 식(11) 및 식(12)의 관계로부터

$$A - \frac{1}{2}[S(x, y) + S(x+1, y)] = \frac{1}{2}[E(x, y) + E(x+1, y)] \quad (13)$$

을 얻을 수 있다. 이를 식(10)에 대입하면  $E(x+0.5, y)$ 는

$$E(x+0.5, y) = \frac{1}{2}[E(x, y) + E(x+1, y)] \quad (14)$$

와 같다. 이 식을 살펴보면 반화소 단위의 탐색점  $(x+0.5, y)$ 에서 오차 블록  $E(x+0.5, y)$ 는 화소 단위의 오차 블록  $E(x, y)$ 와  $E(x+1, y)$ 의 합으로 표현됨을 알 수 있다. 또한 이것은 수평 및 대각선의 각 방향에 대하여 동일하게 적용 될 수 있다. 그러므로 위의 관계식에 삼각 부등식을 적용하여 반화소 단위의 MAD의 최소 범위를 구한 뒤, 이를 이용하여 정합이 필요한 탐색점에 대해서만 블록 정합을 행한다면 정합에 필요한 계산량을 줄임으로써 고속으로 반화소 단위의 움직임 추정을 행할 수 있다.

입의  $N \times N$  행렬 M의  $L_1$  놈 (norm)은

$$\|M\| = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |m_{ij}| \quad (15)$$

와 같이 정의되고, 삼각부등식은

$$| \| \mathbf{M}_1 \| - \| \mathbf{M}_2 \| | \leq \| \mathbf{M}_1 + \mathbf{M}_2 \| \quad (16)$$

와 같이 표현된다. 이를 식(14)의 우변에 적용하면

$$\begin{aligned} & | \frac{1}{2} \| \mathbf{E}(\mathbf{x}, \mathbf{y}) \| - \frac{1}{2} \| \mathbf{E}(\mathbf{x} + 1, \mathbf{y}) \| | \quad (17) \\ & \leq \frac{1}{2} \| \mathbf{E}(\mathbf{x}, \mathbf{y}) + \mathbf{E}(\mathbf{x} + 1, \mathbf{y}) \| \end{aligned}$$

와 같다. 이때, 식(14)로부터  $\| \mathbf{E}(\mathbf{x}, \mathbf{y}) + \mathbf{E}(\mathbf{x} + 1, \mathbf{y}) \| = \| \mathbf{E}(\mathbf{x} + 0.5, \mathbf{y}) \|$ 이므로, 이를 식 (17) 대입하고 양변을  $N^2$ 으로 나누면

$$\begin{aligned} & | \frac{1}{2N^2} \| \mathbf{E}(\mathbf{x}, \mathbf{y}) \| - \frac{1}{2N^2} \| \mathbf{E}(\mathbf{x} + 1, \mathbf{y}) \| | \quad (18) \\ & \leq \frac{1}{2N^2} \| \mathbf{E}(\mathbf{x} + 0.5, \mathbf{y}) \| \end{aligned}$$

와 같이 구할 수 있다. 이 식에서  $\frac{1}{N^2} \| \mathbf{E}(\mathbf{x}, \mathbf{y}) \|$  및  $\frac{1}{N^2} \| \mathbf{E}(\mathbf{x} + 1, \mathbf{y}) \|$ 는 화소 단위의 MAD, 그리고  $\frac{1}{N^2} \| \mathbf{E}(\mathbf{x} + 0.5, \mathbf{y}) \|$ 는 반화소 단위의 MAD를 의미한다. 그러므로 반화소 단위의 MAD의 최소 범위는 인접한 화소 단위의 MAD의 차로 구할 수 있음을 알 수 있다. 이때, 반화소 단위의 MAD의 최소 범위를 구하기 위한 계산량은 화소 단위의 MAD들을 화소 단위의 움직임 추정을 통하여 얻을 수 있으므로 화소 단위의 MAD차를 구하기 위한 몇 번의 뺄셈의 계산량으로 거의 무시 할 수 있다. 그러므로, 이를 이용하여 정합이 필요한 탐색점에 대하여서만 정합을 행한다면 기존의 방법과 같은 성능을 유지하면서도 고속으로 움직임을 추정할 수 있다. 즉 제안한 방법에서는 반화소 단위의 움직임 탐색영역에서는 화소 단위의 움직임 추정 MAD를 이용하여

$$\begin{aligned} & IF(MAD_{Ref} < | \frac{1}{2N^2} \| \mathbf{E}(\mathbf{x}, \mathbf{y}) \| - \frac{1}{2N^2} \| \mathbf{E}(\mathbf{x} + 1, \mathbf{y}) \| | ) \\ & \quad No \ block \ matching \end{aligned} \quad (19)$$

ELSE {

    Block matching

$$IF(MAD(x+0.5, y) < MAD_{Ref},)$$

$$MAD_{Ref} = MAD(x+0.5, y)$$

}

와 같이 정합이 필요한 부분에 대하여서만 블록 정

합을 행한다. 여기서, 기준 MAD인  $MAD_{ref}$ 는 화소 단위의 움직임 추정을 행하여 얻을 수 있는 최소 MAD로서 반화소 단위의 움직임 추정 과정에서 식 (19)에서와 같이 갱신된다. 이와 같은 방법으로 탐색점을 줄임으로써 MPEG-2와 같은 성능을 유지하면서 계산량을 줄일 수 있다. 또한 이것은 수평 및 대각선의 각 방향에 대하여 동일하게 적용 될 수 있다.

제한한 방법의 계산량을 살펴보면, 하나의 탐색점에 대하여 MAD를 구하기 위한 계산량이  $2N^2$ 이므로 한 프레임내에서 기준 MAD가 반화소 단위의 MAD의 최소 범위 보다 큰 탐색점의 총 개수를 P라 하면, 제안한 방법의 총 계산량은

$$C_p = 2PN^2 \quad (20)$$

와 같다. 이 식에서 보는 바와 같이 제안한 방법의 계산량은 기준 MAD가 반화소 단위의 MAD의 최소 범위보다 큰 탐색점의 한 프레임 총 개수 P에 의존하므로 입력 영상의 특성에 따라 약간의 차이는 나타낼 수 있지만, MPEG-2의 방법에 비하여 동일한 추정오차를 가지면서도 많은 계산량을 줄일 수 있다.

#### IV. 실험결과 및 고찰

본 논문에서 제안한 고속 반화소 단위의 블록 정합 알고리즘의 성능 평가를 위하여 다음과 같은 조건하에서 컴퓨터 모의 실험이 행하여졌다. SIF 영상인 FLOWER GARDEN, MOBILE, 그리고 TABLE TENNIS 영상 등이 각각 50 프레임이 사용되었다. 또한 움직임 추정에 사용된 블록의 크기는  $8 \times 8$ 이며, 탐색 범위로는  $-7.5 \sim 7.5$ 이 사용되었다. 그리고 MAD가 정합 척도로 사용되었다. 본 논문에서 제안한 고속 반화소 단위의 블록 정합 알고리즘을 제안 방법 (proposed method), MPEG-2의 반화소 움직임 추정 기법을 기존의 방법 (conventional method)라 부르기로 한다. 본 모의 실험에서는 각 방법들에 대한 결과를

$$PSNR = 10 \log \frac{255^2}{\sigma_e^2} [dB] \quad (21)$$

를 이용하여 평가하였다. 여기서,  $\sigma_e^2$  은 원 영상과 움직임 보상된 영상의 평균 자승 오차이다.

표 1. 제안방법의 평균 PSNR [dB]

Sequences	FSBMA	conventional method	proposed method
FLOWER GARDEN	25.99	27.61	27.61
MOBILE	24.28	25.44	25.44
TABLE TENNIS	30.49	31.46	31.46

표 1은 MOBILE, FLOWER GARDEN 및 FOOTBALL 영상에 대한 50 프레임에 대한 평균 PSNR을 나타내었다. 표 1를 보면 제안한 방법은 MPEG-2에서 사용한 방법과 동일한 성능을 유지함을 알 수 있다. 이는 제안한 방법 1, 2에서는 식(14) 및 식(19)에서 알 수 있듯이 기존의 방법과 동일한 정합 척도의 값을 얻을 수 있고, 반화소 단위의 모든 탐색점에서 정합척도를 비교하여 가장 작은 정합 척도를 가지는 탐색점을 찾는 방법이기 때문이다.

제안한 방법에서는 반화소 단위의 MAD의 최소 범위를 이용하여 필요한 경우에만 블록 정합을 행하여 탐색점 수를 줄임으로써 고속으로 반화소 단위의 움직임 추정을 행하였다. 표 2에는 제안한 방법과 기존의 방법의 50프레임에 대한 평균 계산량을 나타내었다. 이 표를 보면, 제안한 방법의 평균 계산량은 기존의 방법에 비해 FLOWER GARDEN 영상의 경우 36.6%, MOBILE 영상의 경우 32.1%, 그리고 TABLE TENNIS 영상의 경우 30.0% 정도의 감소를 얻을 수 있음을 알 수 있다. 제안한 방법의 계산량은 식(20)에서 알 수 있는바와 같이 입력 영상에 따라 달라질 수 있다. 자세히 살펴보기 위하

표 2. 계산량의 비교

Sequences	conventional method	proposed method
FLOWER GARDEN	100	63.4
MOBILE	100	67.9
TABLE TENNIS	100	70.0

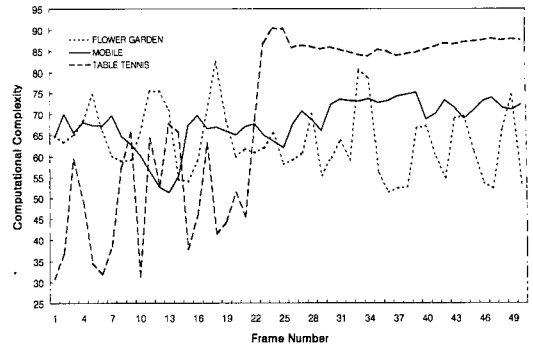


그림 2. FLOWER GARDEN, MOBILE 그리고 TABLE TENNIS 영상에 대한 제안방법의 계산량

여 실험 영상 50프레임에 대한 제안 방법의 계산량을 그림 2에 나타내었다. 이 그림을 보면 계산량의 변화는 단순한 부분이 많은 MOBILE 영상보다 복잡한 부분이 많은 FLOWER GARDEN 및 TABLE TENNIS 영상에서 더 많음을 알 수 있다. 즉, MOBILE 영상의 경우에는 복잡하지 않은 배경이 수평 이동하는 부분이 많으므로 영상에 따른 계산량의 변화가 크지 않음을 알 수 있다. 그러나 TABLE TENNIS 영상에 대한 결과를 보면, 23번째 프레임부터 계산량이 크게 증가함을 볼 수 있는데, 이는 23번째 프레임부터 카메라의 줌 인 (zoom in) 이 일어나기 때문에 반화소 단위 MAD의 최소범위보다 화소단위로 찾은 기준 MAD가 큰 값을 가지게 되어 블록정합을 행하여야 할 블록이 많기 때문이다.

이상의 결과로부터 제안한 방법이 기존의 방법에 비하여 동일한 성능을 유지하면서도 약 30% 이상의 계산량 감소를 얻을 수 있음을 확인할 수 있었다.

## V. 결론

본 논문에서는 반화소 위치에서의 MAD의 최소 범위를 이용한 고속 반화소 단위의 블록 정합 알고리즘을 제안하였다. 제안한 방법에서는 반화소 단위의 MAD의 최소 범위를 화소 단위의 움직임 추정으로 얻어진 MAD들을 이용하여 구한 후, 반화소 단위의 MAD의 최소 범위가 화소 단위의 움직임 추정으로 구한 기준 MAD보다 작은 탐색점에 대해서만 블록 정합을 행하였다. 제안한 방법의 성능을 평가하기 위하여 컴퓨터 모의 실험을 행하였다. 그

결과, 제안한 방법은 움직임 추정 오차측면에서 MPEG-2의 반화소 단위의 움직임 추정 기법과 동일한 성능을 유지하면서 FLOWER GARDEN영상의 경우 36.57%, MOBILE 영상의 경우 32.08%, 그리고 TABLE TENNIS 영상의 경우 29.96% 정도의 계산량을 줄일 수 있음을 확인할 수 있었다.

참 고 문 헌

- [1] ITU-T Recommendation H.261, "Video codec for audiovisual services at p×64 kbits/s."
- [2] ITU-T Recommendation H.263, "Video coding for low bit rate communication."
- [3] ISO/IEC 11172-2, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5Mbits/s: Video."
- [4] ISO/IEC 13818-2, "Information technology - Generic Coding of Moving Pictures and Associated Audio Information: Video."
- [5] A. K. Jain, "Image data compression: A review," Proc. of IEEE, vol. 69, no.3, pp. 349-389, March 1981.
- [6] A. N. Netravali and J. O. Limb, "Picture coding: A review," Proc of IEEE, vol. 68, no. 3, pp. 366-406, March 1980.
- [7] H. G. Musman, p. Pirch, and H. J. Grallert, "Advances in picture coding," Proc. of IEEE, vol. 73, no. 4, pp. 523-548, April 1985.
- [8] K. M. Nam, J. S. Kim, and R. H. Park, "A fast hierarchical motion vector estimation algorithm using mean pyramid," IEEE Trans. Circuits Syst. Video Technol., vol. 5, pp. 344-351, Aug. 1995
- [9] B. Girod, "Motion-compensating prediction with fractional-pel accuracy," IEEE Trans. commun., vol. 41, no. 4, pp. 604-612, April 1993.
- [10] S. L. Iu, "Comparison of motion compensation using different degree of sub-pixel accuracy for interfield/interframe hybrid coding of HDTV image sequences," in 1992 International Conference on Acoustics, Speech and Signal Processing, San Francisco, CA, USA, 1992, vol. 3. pp. 465-468.
- [11] Motion Picture Experts Group, "MPEG Test model 5 draft revision 2," ISO-IEC JTC1/SC29/WG11/602, Nov. 1993
- [12] J. Jeong, and W. Ahn, "Subpixel-accuracy motion estimation using A model for motion compensated error," PCS'93, 13.4. March. 1993.
- [13] Y. Senda, H. Harasaki, and M. Yano. "A Simplified motion estimation using an approximation for the MPEG-2 real-time encoder," ICASSP'95. Vol. 4, pp. 2273-2276, 1995

이 법 기(Bub-Ki Lee)	정회원
통신학회논문지 제24권 제1B호 참조	
정 원 식(Won-Sik Cheong)	정회원
통신학회논문지 제24권 제1B호 참조	
김 덕 규(Duk-Gyoo Kim)	정회원
통신학회논문지 제24권 제1B호 참조	