

# 영상의 국부적 성질을 이용한 가변 크기 블럭 정합 알고리즘

## (A Variable Size Block Matching Algorithm Using Local Characteristics of Images)

金 珍 台\*, 崔 宗 秀\*, 朴 來 弘\*\*

(Jin Tae Kim, Jong Soo Choi, and Rae-Hong Park)

### 要 約

일반적인 BMA는 고정 블럭 크기로 움직임을 찾는다. 저전송률에서 부호화 효율을 증가시키기 위해서는, 배경영역에서는 블럭 크기를 크게 하고 움직임 영역에서는 블럭 크기를 작게할 필요가 있다. 본 논문에서는 가변 블럭 크기의 BMA를 제안한다. 블럭은 영상의 국부적인 평균과 분산에 의해 크기가 결정된다. 시뮬레이션 결과, 제안한 방식은 일반적인 BMA보다 움직임 벡터를 부호화하는데 비트가 절약된다.

### Abstract

The conventional BMA is performed with the fixed block size. For better performance at low bit-rate, the block size is required to be large in relatively stationary area, while small in moving area. Thus, in this paper, a video coding technique using variable block size model is proposed. It decides the block size based on the degree of local motion defined by the local mean and variance of blocks. Computer simulation shows that the proposed method gives comparable performance to the conventional one with less bits required for motion vector coding.

### I. 서 론

영상 부호화(image coding)의 목적은 가능한 한 최소의 비트로 원래의 영상을 충실히 표현하고 재생하는 데 있다. 영상에는 화소(pixel)들의 상관성으로 인해 중복성(redundancy)이 존재하는데 이러한 중복성을 제거함으로써 영상을 표현하는 비트 수를 줄일 수 있다. 이웃하는 화소들간의 공간(spatial) 중복성, 다른 칼라성분에

의한 스펙트럴(spectral) 중복성, 여러 프레임 간의 시간(temporal) 중복성 등이 영상 부호화에 이용된다<sup>[1]</sup>.

TV 신호와 같이 시간축상으로 변하는 영상신호를 부호화하고자 할 때, 데이터의 압축을 위해서는 시간 중복성을 제거하는 것이 필수적이다. 시간 중복성 제거를 이용한 방법이 움직임 보상 부호화(motion compensated coding : MCC)이다. MCC는 크게 움직임을 찾는 단위에 따라서 화소순환법(pel recursive algorithm : PRA)과 블럭정합법(block matching algorithm : BMA)으로 나뉜다<sup>[2]</sup>. 성능면에서는 PRA가 우수하지만 계산량의 문제와 하드웨어의 복잡성 때문에 실제의 시스템에는 BMA가 쓰이고 있다. HDTV 방식(MUSE, HD-MAC, ATV 등) 뿐만 아니라 CCITT의 H.261과 ISO의 MPEG에서도 시간 중복성의 이용과 하드웨어 구현의 용이성 때문에 BMA가 사용되고 있다.

일반적으로 BMA는 입력 영상을 일정한 크기의 블럭

\* 正會員, 中央大學校 電子工學科  
(Dept. of Elec. Eng., Chungang Univ.)

\*\* 正會員, 西江大學校 電子工學科  
(Dept. of Elec. Eng., Sogang Univ.)

接受日字 : 1992年 3月 3日

(※ 본 연구는 상공부 첨단영상기기 개발 사업의 일환으로 생기원 주관하의 위탁연구비에 이루어진 것임.)

으로 나누어 프레임 메모리의 영상과 정합시키는 방법으로 영상내 국부적 움직임(local motion)을 고려하는데 한계가 있다. 영상은 움직임이 큰 부분, 움직임이 작은 부분, 그리고 움직임이 전혀 없는 배경 부분으로 나눌 수 있고, 이러한 영상의 국부적인 움직임의 성질을 BMA에 적용함으로써 BMA의 성능을 향상시키고자 하는 노력들이 시도되어 왔다<sup>[3,5]</sup>. 기존의 가변 블록 정합 알고리즘의 연구는 quad-tree 형태로 큰 블록의 움직임 추정후 예측오차가 크면 블록을 세밀하게 나누는 방식을 취하고 있다. 본 논문에서는 같은 부분에 대해 움직임을 두번이상 추정하는 계산적인 소모량을 피하기 위해 영상 국부의 평균(mean)과 분산(variance)을 이용하여 국부적인 움직임의 정도에 따라 블록의 크기를 가변시켜 BMA의 성능을 향상시키고자 한다.

II. 블록 정합 알고리즘

블록 정합 알고리즘(BMA)은 연속하는 프레임 사이의 움직임을 검출함에 있어 블록 단위로 프레임간의 상관성을 이용하는 방법이다. 즉, BMA는 영상을 고정크기의 블록으로 나누어 그 블록에 해당하는 움직임을 전 프레임의 탐색영역(search region)에서 최대의 상관도를 갖는 블록과 정합시키는 방법이다. 그림 1은 탐색영역과 블록간의 관계를 표시한다. 그림 1에서 t는 현재 프레임을 나타내고, t-τ는 전 프레임을 나타낸다.

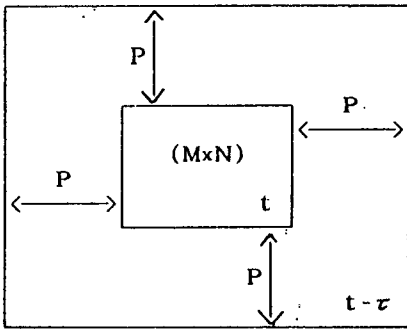


그림 1. 블록과 탐색영역과의 관계  
Fig. 1. Geometry for search region and block.

하는 평가함수(cost function)로는 하드웨어의 용이성을 고려하여 평균절대치오차(mean absolute error ; MAE)가 주로 사용된다. BMA에서 가능한 변위의 수는  $(2P+1)^2$ 이 된다. 모든 변위에 대해서 연산을 하는 방법이 원시적 조사법(brute force search)이다. 원시적 조사법이 갖는 많은 계산량 때문에 평가함수가 quadrant-monotonic하다는 가정하에 변위를 몇개의 단계로 나누어 예상 변위를 순차적으로 줄여 나가면서 계산량을 줄이는 고속 알고리즘이 개발되어 왔다<sup>[7]</sup>. 최근에도 대각 탐색법(cross search algorithm)<sup>[8]</sup>과 병렬(parallel) 처리 기법이 발표되었다<sup>[9]</sup>. 현재 개발된 고속 알고리즘 중에서 성능면과 하드웨어 구현의 용이성 등을 고려하면 Koga 등이 제안한 3단계 조사법(three step search ; TSS)<sup>[10]</sup>이 원시적 조사법에 비해 준최적이라고 할 수 있다. 본 논문에서도 움직임 추정시 3단계 조사법을 사용한다. 또한, 계층적(hierarchical) 움직임 추정 방법<sup>[11]</sup>, 위상 상관(phase correlation)을 이용한 방식<sup>[12]</sup>과 시공간 방정식(spatial-temporal constraint equation)을 이용한 방법<sup>[13]</sup> 등도 연구가 되고 있다.

III. 평균과 분산을 이용한 블록 정합 알고리즘

기존의 BMA는 영상을 고정 크기의 블록으로 나누어 그 블록에 해당하는 움직임 블록을 전 프레임에서 찾는다. 고정 블록 크기의 경우, 영상의 움직임이 거의 없는 배경과 같은 부분에서는 블록의 크기가 작을 필요가 없고, 움직임이 큰 부분에서는 블록의 크기가 크면 섬세한 탐색이 불가능하다. BMA는 한 블록 내의 모든 화소들은 같은 정도의 병진 운동을 한다는 가정하에서 출발하였기 때문에 블록내에 다른 움직임을 갖는 물체가 포함될 경우 BMA 성능을 저하시키는 요인이 된다. 따라서 영상의 국부적인 움직임 정도에 따라 블록의 크기를 가변시키고 그에 따라 움직임 추정 영역을 변화시키면 BMA의 성능을 개선시킬 수 있다.

블록 크기를 가변시키는 기준은 기존의 연구에서는 예측오차를 평가기준으로 하였으나, 본 논문에서는 영상의 공간적 성질을 통계적으로 나타낼 수 있는 평균(mean)과 분산(variance)을 이용한다. K 개의 화소로 이루어진 블록이라 할 때, 블록의 평균( $\bar{X}$ )과 분산( $S^2$ )은 식 (1)과 식(2)에 의한다.

$$\bar{X} = \frac{1}{K} \sum_{i=1,2,\dots,K} X_i \quad (1)$$

$$S^2 = \sum_{i=1,2,\dots,K} (X_i - \bar{X})^2 \quad (2)$$

현 프레임의 블록 크기(MxN)는 HDTV와 같은 시스템에서 16x16 크기가 많이 사용되며(필드의 경우는 16x8), 전 프레임의 탐색영역((M+2P)x(N+2P))에서 최대이동 변위 P는 15가 많이 사용된다<sup>[6]</sup>. 상관도를 측정

가변되는 블록의 모양은 부가 정보(side information) 전송을 고려하여 사각형(rectangle)의 모양으로 제한할 필요가 있다. 그림2는 제안한 방법의 순서도(flowchart)이다.

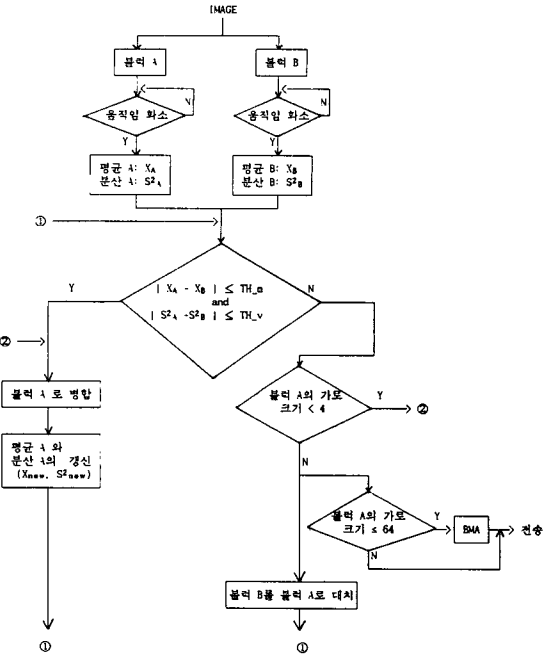


그림 2. 제안한 알고리즘의 순서도  
Fig. 2. Flowchart of the proposed algorithm.

제안한 방법에서는 먼저 두 프레임의 화소간의 차이(difference)의 정도에 따라 영상을 움직임 화소(moving pixel)와 배경 화소(nonmoving pixel)로 구분한다. 블록을 가변시킴으로써 증가하는 부가정보를 제한하기 위해서 가변 블록 한 축의 크기를 고정시켰다. 가변 블록의 세로 크기를 16으로 고정시켰을 경우, 처음 최소 블록(1x16)과 그 다음 최소 블록(1x16)의 움직임 화소에 대해서 각각의 평균과 분산을 구해 그 차이를 비교한다. 그 차의 크기에 따라 두 블록의 merging 여부가 결정된다. 만약 그 차가 임계값 이상이면 두 블록은 독립된 블록으로 처리되어 첫번째 블록에 대해서만 BMA가 수행되고, 두번째 블록은 그 다음에 오는 최소 블록(1x16)과 앞의 과정을 반복한다. 만약 그 차가 임계값 이하이면 두 블록은 같은 속성을 갖는 블록으로 판명되어 merging되어 하나의 블록이 되고 merging된 블록의 평균과 분산은 식(3)과 식(4)와 같이 갱신된다. 여기서, y는 merging되는 블록의 화소 값이고, L은 블록의 차원(dimension)이다.

$$\bar{X}_{new} = \sum (K \bar{X}_{old} + y_i) / (K+1) \tag{3}$$

$$S^2_{new} = \sum S^2_{old} + (y_i + \bar{X})^2 + K(\bar{X}_{new} - \bar{X}_{old})^2 \tag{4}$$

for, i=1,2,...,L

이 merging된 블록은 그 다음에 오는 최소 블록(1x16)과 다시 merging 여부 판정 루틴(routine)을 거친다. 이러한 과정을 반복하여 전체 영상을 처리하면, 영상은 (1x16)부터 (영상의 가로 크기 x 16)까지의 다양한 크기의 블록으로 나뉘어 진다. 이 때 BMA의 효율성과 가변 블록 크기를 나타내는 부가정보량 등을 고려하여 가장 작은 가변 블록 크기를 (4x16)으로 한다. 또한 가변 블록의 가로 길이가 64를 넘으면 움직임이 없는 블록으로 보아 BMA 과정을 생략하고 움직임 벡터를 (0,0)로 한다. 만일 (0,0)로 하는 경우 예측오차가 크면 64 크기이상의 가변 블록도 움직임 추정을 한다. 계산량의 절감을 위해서 가변 블록의 가로 길이가 16이상이면 가로·세로 2:1 부표본화(subsampling)하여 BMA를 수행한다. BMA의 평가함수는 MAE를 사용하고, BMA 과정은 3단계 조사법으로 integer pel 단위까지 처리한다. 가변 블록의 가로 크기를 고정시킨 경우도 축의 방향만 바꾸어서 위와 비슷한 과정을 거친다. 제안한 알고리즘은 그림 3과 같다. 그림3에서 merge()는 두 블록의 평균과 분산을 구하는 함수(function)이고, search()는 움직임 벡터를 3단계 조사법으로 구하는 함수이다. 변수 block은 블록의 세로 크기를 나타낸다. search() 함수에서 block이 64보다 크면, 움직임 벡터는 (0,0)로 set된다.

```

for(p=0;p<R_Y:P+=16){
  start=block=0;
  for(num=0:num<R_X:num++){
    merge();
    if(d_mean<=TH_m && d_var<=TH_v);
    else{
      block=num-start+1;
      if(block<4){block=4; num=start+block-1;}
      search();
      start=num+1;
    }
  }
}

```

그림 3. 제안한 알고리즘(세로 크기를 16으로 고정시킨 경우)  
Fig. 3. The proposed algorithm (fixed vertical size 16).

IV. 컴퓨터 시뮬레이션과 검토

대상 영상으로는 HDTV용 영상신호 「Models」와 MPEG(moving picture experts group)지정 표준영상인 「Football」과「Popple」을 사용했다. 「Models」은 움직임의 종류와 크기가 다양한 영상이고, 「Football」은 움직임이 전체적으로 상당히 크고 배경에 고주파 성분을 많이 포함한 영상이며, 「Popple」은 매우 정적이고 극히 일부 영역에서만 규칙적인 회전 움직임이 있는 영상이다. 대상 신호들은 CCIR 601 형식으로 4:2:2의 디지털 신호로 휘도신호 Y와 색차신호 Cb,Cr로 구성되어 있다 (Y·Cb·Y·Cr). 본 시뮬레이션은 휘도신호(Y)만을 대상으로 한다. 세 영상신호 모두 비월주사된 신호이기 때문에 홀수번째 필드만 세로축으로 선형 보간하여 순차주사된 형태로 바꾸어 그것을 한 프레임으로 하였다. 사용된 영상의 크기는 704x480이다. 본 시뮬레이션은 각 영상을 10프레임씩 처리하였으며, MIPS사의 12mips 워크스테이션 RS2030상에서 수행되었다. 그림 4는 본 시뮬레이션에서 사용한 원영상들이다.

본 컴퓨터 시뮬레이션에서 첫번째 프레임은 이미 복구되어 프레임 메모리에 있다고 가정하고 두번째 프레임부터 처리하였다. 움직임 보상 후의 예측오차(prediction error)는 양자화 계단폭(step size) 8을 갖는 midriser형 균일 양자화기로 양자화된다. 제한한 방법은 3개의 임계값이 필요한데, 실험적으로 평균차에 대한 임계값과 다른 임계값들을 유기적으로 결합시켰다. 분산차에 대한 임계값은 평균차에 대한 임계값에 5를 곱하였으며, 움직임 화소를 구별하는 임계값은 나누기 2를 하였다. 그림 5는 가변 블럭의 세로 크기를 16으로 고정된 경우의 각 영상에서 블럭이 나누어진 모양을 나타낸다. 그림 6은 가변 블럭의 가로 길이를 고정된 경우이다.

표 1은 움직임 보상된 영상의 PSNR을 나타낸다. 표 2는 전체 영상에서 나누어진 블럭의 갯수를 표시하고, 표 3은 나누어진 블럭에 대한 entropy이다. 이 때 사용된 식은 다음과 같다. 여기서, MSE는 원영상과 복구된 영상의 평균제곱오차(mean squared error)이고, p(i)는 가변 블럭 크기에 대한 확률이다.

$$PSNR=10 \log_{10}(255^2/MSE) \quad [dB]$$

$$Entropy= - \sum p(i) \log_2 p(i) \quad [bits]$$

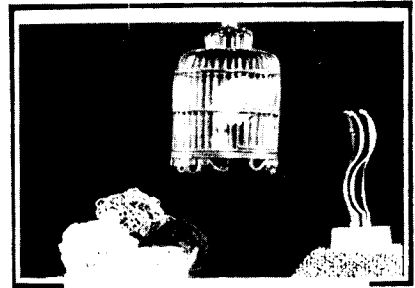
표 1,2,3에서 Algorithm I는 가변 블럭의 세로 크기를 고정시킨 경우이며, Algorithm II는 가변 블럭의 가로 크기 고정의 경우이다. Algorithm III은 비교를 위해서 고정 블럭 크기(16x16)의 BMA를 수행시킨 경우이다. BMA는 TSS를 사용한다.



(a) Models



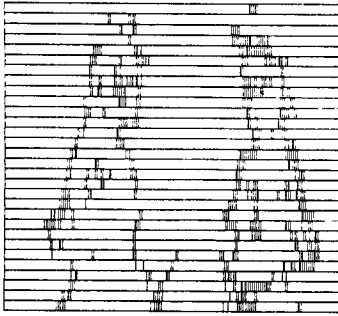
(b) Football



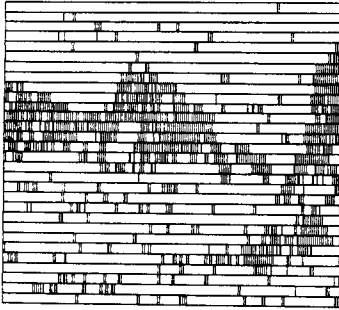
(c) Popple

그림 4. 시뮬레이션에 사용된 영상  
Fig. 4. Original images.

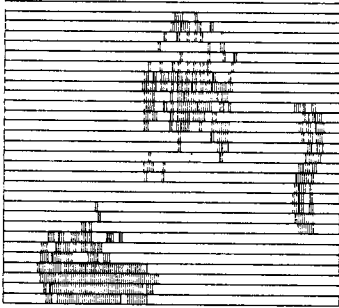
표 1은 움직임 보상 후의 복구된 영상의 PSNR을 나타낸다. 이때 예측오차에 대한 부호화는 고려하지 않았으며, 다음 프레임의 움직임 추정을 위해 예측오차를 3가지 방법 모두에 동등한 조건하의 균일 양자화기를 적용하여 처리하였다. PSNR 측면에서 「Models」 영상에서는 고정 블럭 크기의 BMA 경우가 약 1 dB 정도 성능이 우수하고, 「Popple」 영상에서는 가변 블럭 크기의 BMA가 약 0.5dB정도 우수하며, 「Football」 영상은 거의 비슷한 성능을 보인다. 따라서 배경영역에서는 블럭의 크기를 크게 하여도 화질에 영향을 주지 않으며, 움직임 영역에서는 블럭의 크기를 작게하면 화질이 개선된다. 나누어지는 가



(a)Models



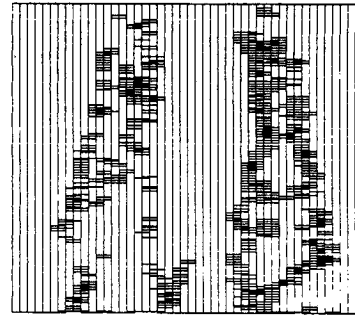
(b) Football



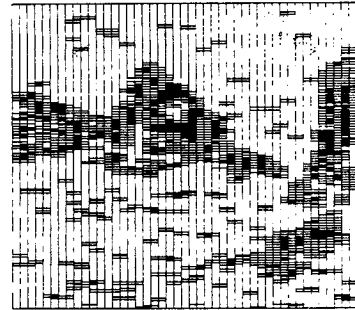
(c) Popple

그림 5. 세로 크기를 고정한 경우의 크기 블록 형태  
 Fig. 5. Variable size block types with the vertical size fixed.

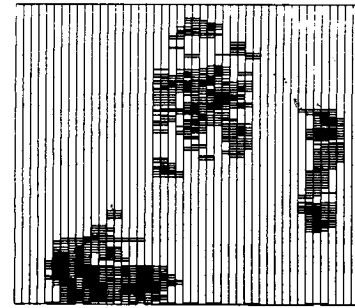
변 블록의 수를 증가시키면 PSNR도 같이 증가하지만 전체적인 화질을 고정 블록 크기의 경우와 비슷하게 유지되도록 임계값을 조정하였다. 사용된 임계값은 모든 영상, 모든 프레임에 같은 값을 적용한다. 표2에서 보면, BMA를 수행하는 블록의 갯수는 고정 블록 크기의 경우 프레임당 1320개로 구성되고, 이에 대한 움직임 벡터(motion vector)의 전송이 필요하다. 가변 블록 크기의 경우에 블



(a)Models



(b) Football



(c) Popple

그림 6. 가로크기를 고정한 경우의 가변 크기 블록 형태  
 Fig. 6. Variable size types with the horizontal size fixed.

록의 갯수는 고정 블록 크기의 경우에 비하여 「Football」 영상은 약 100개, 「Models」과 「Popple」 영상은 약 550개 정도가 적게 구성되어 있다. 표 3에서, 가변 크기 BMA의 경우에는 블록 크기에 대한 부가정보가 필요한데 본 시뮬레이션에서는 나타난 부가정보의 엔트로피(entropy)는 가변 블록의 수직 크기 고정의 경우 「Models」 영상은 평균 3.478bits/block, 「Football」은

표 1. 움직임 보상된 영상의 PSNR.  
Table 1. PSNR of motion compensated images.

	(a) Models (dB)		
	Algorithm I	Algorithm II	Algorithm III
Model 2	26.48	26.54	27.83
Model 3	25.79	26.12	27.22
Model 4	25.06	25.46	26.67
Model 5	25.09	25.81	26.61
Model 6	25.54	26.02	26.56
Model 7	25.47	25.80	26.73
Model 8	25.36	26.00	26.76
Model 9	25.71	25.95	27.05
Model 10	26.36	26.85	27.04
평균	25.65	26.06	26.98

	(b) Football (dB)		
	Algorithm I	Algorithm II	Algorithm III
Football 2	23.62	23.69	24.23
Football 3	24.28	24.21	24.27
Football 4	24.58	24.47	24.17
Football 5	23.78	23.94	23.83
Football 6	23.97	23.95	24.23
Football 7	23.75	23.63	23.90
Football 8	24.03	24.12	23.83
Football 9	23.94	23.77	23.64
Football 10	23.23	23.29	23.48
평균	23.91	23.90	23.95

	(c) Popple (dB)		
	Algorithm I	Algorithm II	Algorithm III
Popple 2	27.28	26.88	26.96
Popple 3	26.59	26.19	25.80
Popple 4	25.91	25.42	25.13
Popple 5	26.01	25.70	25.42
Popple 6	26.42	26.01	26.04
Popple 7	26.48	26.09	25.75
Popple 8	26.39	25.85	25.93
Popple 9	26.25	26.10	25.87
Popple 10	26.53	26.34	26.22
평균	26.43	26.06	25.90

표 2. 블록의 개수  
Table 2. Number of blocks.

	(a) Models (수/프레임)		
	Algorithm I	Algorithm II	Algorithm III
Model 2	644	653	1320
Model 3	690	683	1320
Model 4	721	723	1320
Model 5	770	759	1320
Model 6	855	838	1320
Model 7	846	871	1320
Model 8	835	833	1320
Model 9	816	792	1320
Model 10	807	780	1320
평균	776	770	1320

	(b) Football (수/프레임)		
	Algorithm I	Algorithm II	Algorithm III
Football 2	1183	1236	1320
Football 3	1057	1106	1320
Football 4	1078	1148	1320
Football 5	1114	1182	1320
Football 6	1180	1192	1320
Football 7	1191	1227	1320
Football 8	1275	1269	1320
Football 9	1331	1384	1320
Football 10	1450	1510	1320
평균	1205	1250	1320

	(c) Popple (수/프레임)		
	Algorithm I	Algorithm II	Algorithm III
Popple 2	744	727	1320
Popple 3	779	735	1320
Popple 4	761	712	1320
Popple 5	763	741	1320
Popple 6	775	742	1320
Popple 7	765	754	1320
Popple 8	788	761	1320
Popple 9	771	788	1320
Popple 10	785	782	1320
평균	770	749	1320

2.234bits/block, 「Popple」은 2.229bits/block이고, 가변 블록의 수평 크기 고정화 경우도 비슷한 정도이다. 따라서 한 프레임의 부가정보의 양은 「Models」 영상의 경우 평균 2699bits/frame, 「Football」은 평균 2692bits/frame, 「Popple」은 평균 1716bits/frame 이 되며, 즉 0.005에서 0.009bits/pixel 정도이다. 그러

나 블록의 수가 고정크기의 경우보다 평균적으로 프레임 당 100개에서 550개 적기 때문에, 움직임 벡터 전송시 비트율을 절감할 수 있다. 또한 가변 블록의 크기가 64 이상이면, 배경영역으로 처리되기 때문에 블록의 크기에 대한 정보만이 필요하므로 전체적인 비트율은 고정 블록 크기의 BMA 방법보다 비트율을 크게 낮출 수 있다.

표 3. 블록 크기의 entropy.

Table 3. Entropy of the block size.

(a) Models (bits/block)

	Algorithm I	Algorithm II	Algorithm III
Model 2	3.574	3.433	0
Model 3	3.600	3.388	0
Model 4	3.527	3.354	0
Model 5	3.526	3.542	0
Model 6	3.375	3.468	0
Model 7	3.528	3.323	0
Model 8	3.412	3.421	0
Model 9	3.435	3.557	0
Model 10	3.325	3.571	0
평균	3.478	3.429	0

(b) Football (bits/block)

	Algorithm I	Algorithm II	Algorithm III
Football 2	2.747	2.853	0
Football 3	2.232	2.368	0
Football 4	2.251	2.419	0
Football 5	2.104	2.272	0
Football 6	2.169	2.369	0
Football 7	2.086	2.331	0
Football 8	2.050	2.392	0
Football 9	2.140	2.280	0
Football 10	2.324	2.417	0
평균	2.234	2.411	0

(c) Popple (dB)

	Algorithm I	Algorithm II	Algorithm III
Popple 2	2.272	2.345	0
Popple 3	2.268	2.372	0
Popple 4	2.237	2.325	0
Popple 5	2.246	2.473	0
Popple 6	2.147	2.285	0
Popple 7	2.222	2.364	0
Popple 8	2.202	2.403	0
Popple 9	2.264	2.325	0
Popple 10	2.204	2.340	0
평균	2.229	2.359	0

가변 블록 알고리즘과의 비교에서 가변 블록의 가장 기본적인 형태는 quad-tree와 binary-tree라 말할 수 있다. 나누어지는 기준(임계값과 블록의 크기)에 따라 블록의 갯수와 영상의 화질이 결정되지만, 블록의 갯수를 비슷한 조건하에서 quad-tree 방법을 처리한 결과 제안한 방법과 비슷한 결과를 보인다.

## V. 결 론

일반적인 BMA 방법은 영상을 고정 크기 블록으로 나누어 움직임 추정을 행하고 있기 때문에 영상의 국부적인 움직임의 정도를 고려하는 데는 한계가 있다. 배경 영역에서는 BMA를 수행하는데 불필요한 계산량을 소모하고 있으며, 움직임이 작고 빠른 경우는 예측효율이 떨어진다.

제안한 알고리즘은 영상 국부의 평균과 분산을 이용하여 블록 크기를 가변시켜 BMA를 수행하였다. 가변된 블록의 갯수는 고정 크기 BMA보다 적은 수로 구성되며 성능은 비슷한 정도이다. 나누어진 블록에 대한 부가 정보도 크지 않다. 따라서 제안한 방법은 기존의 BMA 방법보다 낮은 비트율에서 비슷한 성능을 보였다. 임계값 조절(buffer control을 사용할 경우)에 의해 같은 비트율에서 성능향상을 이룰 수 있다.

앞으로의 연구는 영상에 따라 임계값을 선택적으로 결정하는 방법과 예측 오차를 부호화 할 때 효율적인 형태의 블록 크기가 되도록 하는 방법의 모색과 새로운 배경이 나타나거나(uncovered background) 새로운 물체가 영상내로 들어오는 경우 작은 블록 크기로 BMA를 수행해도 상대적인 예측오차가 증가하는데, 이때 프레임내 부호화(intraframe coding) 방법과의 결합이 필요하리라 본다.

## 감사의 글

본 논문의 완성을 위해 많은 조언과 지원을 해주신 생산기술연구원 HDTV 개발사업단의 심영석 박사님과 관계자 여러분께 깊은 감사를 드립니다.

## 參 考 文 獻

- [1] M. Rabbani and P.W. Jones, *Digital Image Compression Techniques*, SPIE Optical engineering Press, 1991.
- [2] H.G. Musmann, P. Pirsch, and H.J. Grallert, "Advanced in picture coding," *Proc. IEEE*, vol. 73, no. 4, pp. 523-548. 1985.
- [3] A. Puri, H.M. Hang, and D.L. Schilling, "Interframe coding with variable block size motion compensation," *Proc. GLOBE-COM'87*, pp. 271-275, 1987.
- [4] T.B. Yu, M.H. Chan, and A.G. Constantinides, "Low bit rate video coding using variable block size model," *Proc. ICASSP'90*, pp. 2229-2232, 1990.

- [5] G.J. Sullivan and R.L. Baker, "Rate-distortion optimized motion compensation for video compression using fixed or variable size block," *Proc. GLOBECOM'91*, pp. 85-90, 1991.
- [6] F.W.P. Vreeswijk, W. Jonker, J.R.G. Leenen, and J. van der Meer, "An HD-MAC coding system," *Proc. of the 2nd International Workshop on Signal Processing of HDTV*, pp. 137-143, 1988.
- [7] 김진태, 주창희, 최중수, "움직임 추정을 위한 Block Matching Algorithm의 비교," 대한전자공학회 텔레콤, vol. 4, no. 2, pp. 27-33, 1988.
- [8] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol. COM-38, no. 7, pp. 950-953, 1990.
- [9] L.G. Chen, W.T. Chen, Y.S. Jehng, and T.D. Chiueh, "An efficient parallel motion estimation for digital image procession," *IEEE Trans. Circuits Syst. for VT*, vol. 1, no. 4, pp. 378-385, 1992.
- [10] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," *Proc. NTC'81*, pp. G5.3.1-5.3.5, 1981.
- [11] J.W. Woods and t. Naveen, "Motion compensated multiresolution of HDTV," *Proc. GLOBECOM'91*, pp. 77-79, 1991.
- [12] M. Ziegler, "Hierarchical motion estimation using the phase correlation method in 140 Mbit/s HDTV coding," *Proc. of the 3rd International Workshop on HDTV*, pp. 77-83, 1989.
- [13] J.S. Lim, *Two-Dimensional Signal and Image Processing*, Prentice-Hall, Inc., pp. 497-507, 1990.

---

 著 者 紹 介
 

---

金 珍 台 (正會員) 第 26 券 第 9 號 參照  
 현재 중앙대학교전자공학과  
 박사과정

朴 來 弘 (正會員) 第 23 券 第 6 號 參照  
 현재 서강대학교 전자공학과  
 부교수

●

崔 宗 秀 (正會員) 第 29 券 B編 第 1 號 參照  
 현재 중앙대학교 전자공학과 교수