

# 움직임 추정 오차 분포를 이용한 H.263 부호화기의 진보 예측 모드의 동적 선택

## (A Dynamic Selection of Advanced Prediction Mode in H.263 Encoder Using Error Distribution of Motion Estimation)

許泰元 \* , 李根泳 \*

(Tae-Won Hur and Keun-Young Lee)

### 요 약

본 논문은 H.263 부호화기에서 진보 예측 모드(Advanced Prediction Mode, APM)를 동적으로 적용하여 계산량을 줄이고 부호화 효율을 높일 수 있는 APM의 동적 선택(Dynamic APM, DAPM)방법을 제안하였다. 화면 내에 존재하는 움직임 성분을 분석하고, 그 성격에 따라 기본 예측 모드(Default Prediction Mode, DPM)와 APM을 동적으로 적용할 수 있도록 하였다. 이를 위하여 매크로 블록 단위로 추정된 각 탐색점에서의 예측 오차로 구성된 움직임 추정 오차 분포(Error Distribution of Motion Estimation, EDME)를 구성하고, 이 분포를 네 개의 영역으로 분할한 다음, 각 분할 영역의 최소 예측 오차와 매크로 블록 단위로 추정된 예측 오차와의 비를 구하여, 블록 단위의 움직임 추정을 수행할 것인가를 결정하였다. 선형 움직임이 대부분인 화면에 대해서는 DPM으로 부호화 하도록 하여, 부분적으로 예측 효율이 떨어지는 것을 막을 수 있었으며, 고정된 APM(Fixed APM, FAPM)으로 부호화 하는 경우에 발생하는 계산량을 30%정도로 줄이면서, 평균적으로 FAPM으로 부호화 된 화질과 비슷한 결과를 얻을 수 있었다.

### Abstract

In this paper, we proposed a dynamic selection scheme of advanced prediction mode(DAPM), which reduces computational cost and improves coding efficiency. We can select the mode between default prediction mode(DPM) and advanced prediction mode(APM) according to motion components in a frame dynamically. For this purpose, we defined error distribution of motion estimation (EDME) as sum of absolute difference(SAD) for each searching points. This distribution region is divided to four subregions. We calculate minimum values in each subregions and then, we determine whether block motion estimation is performed or not depending on the results. As a result, we reduced computational complexity to 30% without degradation of image quality compared to fixed APM(FAPM) by selecting DPM for linear movement.

### I. 서 론

효율적인 비디오 부호화 방법은 영상의 공간적, 시

간적 중복성을 제거하는데 근간을 두고 있다. 공간적 인 중복성은 이산 여현 변환이나 웨이브렛 변환과 같은 변환 방법에 의하여 그 중복성이 제거될 수 있으며, 일반적으로, 시간적인 중복성은 움직임 보상에 의하여 다음 화면을 예측함으로써 제거될 수 있다. 일반적으로, 이러한 움직임 추정 방법에는 전역 탐색

\* 正會員, 成均館大學校 電子工學科

(Dept. of Electronics, Sungkyunkwan Univ.)

接受日字: 1997年11月17日, 수정완료일: 1998年5月7日

(Full Search), 3단계 탐색(Three Step Search, TSS)방법 등에서 많이 이용되는 블록 정합 알고리즘<sup>[1]</sup>이 이용되고 있다. 블록 정합 알고리즘은 MPEG<sup>[2]</sup>, H.261<sup>[3]</sup>, H.263<sup>[4]</sup> 등의 동영상 표준화에 널리 채택되어 왔다. 하지만, 블록 정합 알고리즘은 움직임 추정/보상에 있어 블록 단위로 수행되기 때문에 복원된 영상에 블록화 현상을 수반한다. 특히, 저 전송율로 부호화 하는데 있어 이러한 블록화 현상은 주관적인 화질을 열화 시킨다. 이러한 블록화 현상을 줄이기 위하여 많은 방법들이 연구되어 왔는데, 블록 내에 다양한 움직임 성분이 포함되어 있을 경우는 블록을 분할하고, 각각의 분할된 블록의 움직임을 나누어서 추정하는 방법과 움직임 보상시 이웃하는 블록의 움직임을 반영하는 중복 움직임 보상 방법이 그 대표적인 예이다. H.263 권고안에서는 이러한 방법을 선택사항으로 마련하고 있다.

H.263 권고안은 부호화 성능을 개선할 수 있는 네 가지 선택사항을 마련해 놓고 있는데, 먼저 비제한 움직임 벡터 모드(Unrestricted Motion Vector mode, UMVM)<sup>[5]</sup>는 움직임 벡터 성분을 화면 내에 한정하는 제한을 없앤 것이다. 또한 움직임 벡터의 범위를 확장하여 기본 예측 모드(Default Prediction mode, DPM)에 비해 큰 움직임 벡터를 허용하였다. UMVM은 카메라의 움직임이 있는 경우에 부호화 효율을 높인다. 두 번째로 구문 기반 산술 부호화 모드(Syntax-based Arithmetic Coding mode)를 선택사항으로 두었는데, DPM의 허프만 부호화에 비해 적은 비트로 비슷한 화질을 얻을 수 있다. 세 번째는 진보 예측 모드(Advanced Prediction mode, APM)로, 블록 단위로 움직임 추정을 수행함으로써, 각 매크로 블록당 4개의 움직임 벡터를 전송하는 방법이다. 또한 APM은 중복 움직임 보상을 수행하여 낮은 전송률에서 발생하는 블록화 현상을 줄일 수 있다. 마지막으로, PB화면 모드(PB-frames mode)는 두 개의 화면을 하나의 단위로 부호화 하는 방법이다. P(Predictive)화면은 이전 화면으로부터 움직임 추정/보상되어 부호화 되고, P 화면 사이의 B(Bidirectional)화면은 앞뒤의 P화면을 이용해 보간하여 부호화 된다. 네 가지 선택사항은 단독으로 적용될 수도 있고, 복합적으로 선택되어 사용될 수 있다.

APM으로 부호화 되면, 중복 움직임 보상 방법으로 블록이 복원되며, 움직임 추정시에는 블록 단위의 움

직임을 추정하게 된다. 일반적으로, 중복 움직임 보상 방법은 영상의 움직임이 매크로 블록내에서 여러 움직임이 비선형적으로 존재하는 경우에 움직임 보상된 예측 오차를 감소시키는 것으로 알려져 있다. 하지만, 움직임 추정시 블록단위의 움직임 추정과 중복 움직임 보상을 위한 추가적인 계산이 필요하게 된다. DPM으로 부호화 하는데 비해 움직임 추정시 1.25배 정도의 계산량과 움직임 보상시 화소당 네 번의 곱셈과 두 번의 덧셈을 추가적으로 하게 된다. 또한, 화면 내에 선형적인 움직임만이 존재하는 경우에는, 이웃하는 매크로 블록의 화소 값을 반영하는 중복 움직임 보상을 하기 때문에 예측된 효율을 떨어뜨릴 수도 있다.

본 논문에서는 FAPM으로 부호화 하는 과정에서 부분적으로 예측 효율이 저하되는 것을 막고 추가적으로 발생하는 계산량을 줄일 수 있는 APM의 동적 선택을 위한 부호화 방법을 제안한다. 화면 내에 존재하는 움직임 성분을 매크로 블록 단위로 분석하여 전체 화면에 존재하는 움직임 성분이 선형적이라면, DPM으로 부호화하고, 그렇지 않은 경우에는 APM으로 부호화 하도록 한다. 결과적으로, 중복 움직임 보상 과정이 적용됨으로써 발생하는 부호화 효율의 저하를 방지할 수 있다. 또한 매크로 블록의 움직임 성분이 선형으로 분류되면, 블록 단위의 움직임 추정 과정을 생략하여, FAPM에 비하여 부호화 효율을 높일 수 있다.

II장에는 APM에 대한 일반적인 사항이 고찰되었으며, III장에는 제안한 부호화기의 동작을 기술하였으며, IV장에는 실험 결과와 고찰 사항을 기술하였으며, V장에는 앞으로의 연구 과제와 제안한 부호화기가 효율적으로 적용될 수 있는 분야에 대하여 고찰하였다.

## II. 진보 예측 모드(Advanced Prediction Mode, APM)

이 모드를 사용하여 부호화 하는 경우에는 중복 움직임 보상이 사용된다는 것과 매크로 블록당 4개의 움직임 벡터를 가지는 것으로 특징 지을 수 있다. 또한 UMVM의 화면의 경계를 넘어서는 움직임 성분에 대하여도 움직임 추정이 이루어진다. UMVM의 움직임 벡터 범위 확장은 이루어지지 않는다. 만일 APM이 PB화면 모드와 함께 사용된다면, 중복 움직임 벡터 보상은 P화면에만 적용되고 B화면에는 적용되지 않는다. 다음은 APM에 적용되는 두 가지 사항에 대하여

살펴보겠다.

1. 블록 단위의 움직임 추정

APM에서 매크로 블록당 움직임 벡터와 블록별 움직임 벡터의 사용여부는 각 매크로 블록의 부호화 블록 형식(Macroblock type & Coded Block Pattern for Chrominance, MCBPC) 부호어에 의하여 표시된다. 만일 하나의 움직임 벡터가 사용된다면, 이것은 네 개의 움직임 벡터가 모두 같음을 의미한다. 만일 MCBPC가 블록별 움직임 벡터를 의미한다면, 첫 번째 움직임 벡터는 움직임 벡터 부호어 (Motion Vector Data, MVD)에 의해 전송되고 부가적인 세 개의 움직임 벡터가 더 전송된다. 움직임 벡터는 추정된 움직임 벡터 값과 예측기에서 구해진 예측 값의 차이가 부호화 된다. 움직임 벡터의 예측은 매크로 블록에 대한 움직임 벡터인 경우에는 매크로 블록의 왼쪽, 위쪽, 위쪽의 오른쪽 매크로 블록들의 움직임 벡터들의 중간값을 구하여 이루어진다. 또한 블록별로 움직임 추정이 이루어진 경우의 움직임 벡터의 예측은 그림 1과 같이 현재 블록의 움직임 벡터  $MV$ 의 예측은  $MV_1, MV_2, MV_3$ 의 중간값으로 구해진다. 색차 성분 블록의 움직임 벡터는 네 개의 움직임 벡터의 합을 8로 나눈 값을 사용한다.

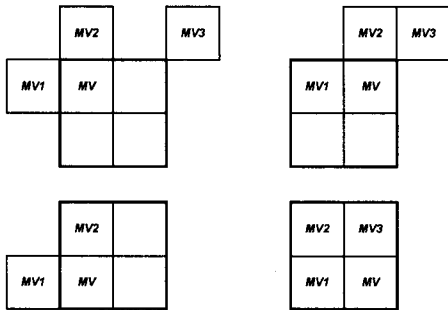


그림 1. 매크로 블록내의 각각의 휘도 성분 블록에 대한 예측 후보 벡터

Fig. 1. Candidate predictors of luminance blocks within a macroblock.

2. 중복 움직임 보상

그림 2와 같이 중복 움직임 보상은 블록단위로 이루어지며,  $8 \times 8$  블록내의 각각의 화소는 세 개의 예측 값의 가중평균을 이용한다. 각 참조 화면의 블록의 기호에 대하여  $l, r, t, b$ 는 왼쪽, 오른쪽, 위, 아래에

인접한 블록을 나타낸다. 세 개의 예측 값은 현재의 휘도 블록의 움직임 벡터  $MV_C$ 와 이웃하는 블록의 움직임 벡터  $MV_L, MV_R, MV_T, MV_B$ 중 두 개의 벡터를 사용하여 예측된다. 즉,  $8 \times 8$  블록을 그림2와 같이  $a, b, c, a$ 의 네 부분으로 분할하면,  $a$ 부분 화소의 예측은 참조화면의  $ap, at, ai$  부분의 화소 값의 가중평균이 사용되고,  $b$ 부분 화소의 예측은 참조화면의  $bp, bt, br$  부분의 화소 값의 가중평균이 사용된다. 여기서,  $ap, bp$ 는 현재 블록의 움직임 벡터  $MV_C$ 로 예측되는 부분이며,  $at, bt$ 는  $MV_T$ 로 예측되고  $ai$ 과  $br$ 은 각각  $MV_L$ 과  $MV_R$ 로 예측되는 부분이다.  $c, a$ 부분에 대하여도 같은 방식으로 적용된다. 따라서, 블록내의 각 화소에 대하여 세 개의 예측 화소  $P_0, P_1, P_2$ 는 다음과 같이 표현될 수 있다.

$$\begin{aligned}
 P_0(x, y) &= p(x + MV_{Cx}, y + MV_{Cy}) \\
 P_1(x, y) &= p(x + MV_{Tx}, y + MV_{Ty}), \text{ 또는 } p(x + MV_{Bx}, y + MV_{By}) \\
 P_2(x, y) &= p(x + MV_{Lx}, y + MV_{Ly}), \text{ 또는 } p(x + MV_{Rx}, y + MV_{Ry})
 \end{aligned}
 \tag{1}$$

여기서,  $p(x, y)$ 는 참조 화면에서의 화소 값을 의미하며, 반화소 단위로 예측된다.  $P_0$ 는 현재 블록의 움직임 벡터에 대한 예측 값이며,  $P_1$ 은 위 또는 아래 위치에 인접한 블록의 움직임 벡터에 대한 예측 값이며,  $P_2$ 는 왼쪽 또는 오른쪽의 움직임 벡터에 대한 예측 값이다. 각 움직임 벡터의 선택은  $a, b, c, a$ 로 분할된 영역에 인접한 이웃하는 블록의 움직임 벡터가 사용된다. 각 움직임 벡터의  $x, y$ 기호는 수직, 수평성분을 나타낸다. 블록내의  $(x, y)$ 위치의 예측 화소값  $P(x, y)$ 는  $P_0, P_1, P_2$ 의 가중 평균으로 구해지며, 식 2에서와 같이 구해진다.

$$P(x, y) = (P_0(x, y) \times H_0(x, y) + P_1(x, y) \times H_1(x, y) + P_2(x, y) \times H_2(x, y) + 4) / 8
 \tag{2}$$

여기서,  $H_0(x, y), H_1(x, y), H_2(x, y)$ 는 예측값들에 대한 가중값을 나타내며, 그림 3과 같은 값이 사용된다.  $P_0(x, y)$ 에 대하여는  $H_0$ 의 전체 값이 사용되며,  $P_1(x, y)$ 와  $P_2(x, y)$ 에 대해서는  $H_1$ 과  $H_2$ 의 절반이 이용되는데,  $MV_T$ 가 사용된 경우에는  $H_1$ 의 아래 절반이 사용되며,  $MV_B$ 가 사용된 경우에는  $H_1$ 의 위 절반이 사용된다. 또한  $MV_L$ 이 사용된 경우는  $H_2$ 의 오른쪽 절반이,  $MV_R$ 이 사용된 경우는  $H_2$ 의 왼쪽 절반이 가중치로 이용된다.

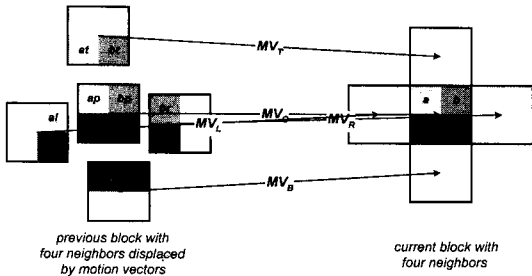


그림 2. 중복 움직임 보상  
Fig. 2. Overlapped block motion compensation

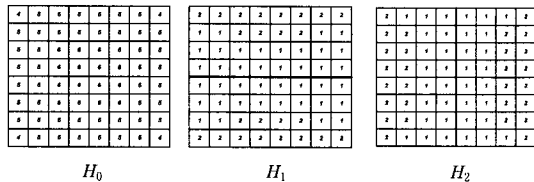
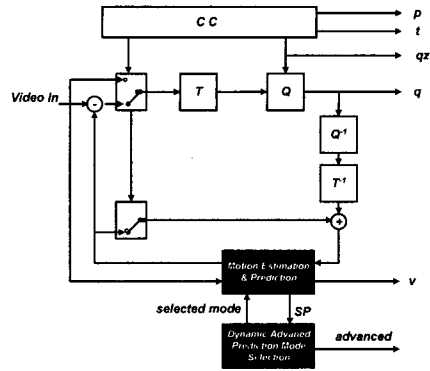


그림 3. 중복 움직임 보상을 위해 사용된 가중치 행렬  
Fig. 3. Weighting matrix for overlapped motion compensation.



- T : 변환
- Q : 양자화기
- CC : 부호화 제어
- p : 인트라/인터
- t : 전송 여부
- qz : Quantizer indication
- q : 변환계수에 사용된 양자화 지수
- v : 움직임 벡터
- SP : 블록 단위 움직임 추정
- advanced : 기본/진보 예측 모드

그림 4. 제안한 부호화 방법의 블록 다이어그램  
Fig. 4. Block diagram of proposed coding scheme.

### III. 제안한 동적 진보 예측 모드(Dynamic Advanced Prediction Mode)부호화기

제안한 부호화기의 블록 다이어그램이 그림 4에 나타나 있다. 그림 4에서 H.263의 기본적인 부호화기와 달라진 점은 움직임 추정 부분과 동적 진보 예측 모드(Dynamic Advanced Prediction Mode, DAPM)선택 블록이 추가되었다는 점이다. 기본적인 부호화기는 APM으로 부호화 되는 경우에 움직임 추정시 매크로 블록과 블록 단위로 각각 수행하여 예측 오차를 작게 하는 방법으로 움직임 벡터를 부호화 한다. 그러나, 제안한 부호화기는 움직임 추정이 블록 단위의 움직임 추정 과정을 생략될 수 있도록 구성하였다. 또한 기본적인 부호화기에서 선택된 모드는 전체 비디오 열에 적용되어 부호화 하는데 비하여, 제안한 부호화기는 영상의 특성에 따라 APM을 동적으로 선택하여 부호화 할 수 있도록 설계되었다.

#### 1. 매크로 블록 내의 움직임 분석

##### 1) 움직임 추정 오차 분포 (Error Distribution of Motion Estimation, EDME)

블록 정합 알고리즘에 의하여 매크로 블록 단위로 움직임 추정을 하는 경우, 매크로 블록 내에 복합적인

고 비선형적인 움직임이 존재하는 경우에는 예측 오차를 크게 한다. 이런 경우에는 블록 단위로 움직임을 추정하여 예측 오차를 낮출 수 있다. 그림 5는 이런 경우를 간단한 예를 통하여 도시한 경우이다. 현재 움직임 추정하고자 하는 매크로 블록의 영상이 그림 5(b)와 같고 참조 화면의 탐색 영역의 영상이 그림 5(a)와 같이 주어진다면, 정확한 움직임을 매크로 블록 단위로 추정할 수 있다. 하지만, 탐색 영역의 영상이 그림 5(c)와 같은 경우는 매크로 블록 단위의 움직임 추정은 정확한 움직임을 추정할 수 없고 예측 오차는 큰 값을 가지게 된다. 이 경우는 현재 움직임 추정하고자 하는 매크로 블록 내에 복합적인 움직임이 존재하는 경우이며, 블록 단위로 움직임을 추정하게 된다면, 예측 오차를 낮출 수 있게 된다. 이와 같이 블록 내에 포함되어 있는 움직임 성분의 특성을 분석하기 위하여, 움직임 추정 오차 분포(Error Distribution of Motion Estimation, EDME)를 정의하였다. EDME는 탐색 영역 안에서 매크로 블록 단위의 움직임 추정이 이루어지는 과정에서 발생하는 각 탐색점에서의 예측 오차들로 구성된다. 예측 오차는 절대 오차의 합(Sum of Absolute Difference, SAD)으로 구해진다. EDME의  $x$ 축과  $y$ 축은  $-15 \leq x, y \leq 15$ 의 탐색 영역의 좌표를 의미하며  $z$ 축은 그 좌표에서 계산되는

움직임 추정 오차를 나타낸다. 그림 5(d), (e)는 각각 탐색 영역의 영상이 그림 5(a), (c)와 같은 경우의 EDME를 나타내는데 선형적인 움직임이 존재하는 경우에는 그림 5(d)와 같이 움직임 추정 오차 값의 최소 값이 다른 위치의 값과 뚜렷하게 구분되어 나타난다. 반면에 복합적인 움직임이 존재하는 경우의 그림 5(e)는 최소값이 다른 위치의 값들과 유사한 값을 갖는다. 그림 5(a)와 같은 선형 움직임의 탐색 영역에 대하여 부호화 하는 경우 해당 매크로 블록은, DPM으로 부호화 하면 26비트가 필요하지만, APM으로 부호화 하면 159비트로 늘어난다. 또한, 그림 5(c)의 비선형 움직임 탐색영역의 경우에는 DPM의 경우 360비트, APM의 경우는 290비트가 소요된다. 이러한 사실은 선형 움직임이 존재하는 경우에는 APM으로 부호화 하면 DPM에 비해 부호화 효율이 떨어질 수도 있음을 반영한다. 반면에, 비선형 움직임이 존재하는 경우에는 APM이 DPM에 비해 부호화 효율을 향상시킬 수 있음을 나타낸다. 따라서, 화면 내에 존재하는 움직임 성분의 성격에 따라 DPM과 APM을 동적으로 적용한다면, 부호화 효율을 향상시킬 수 있을 것이다. 다음으로, 이러한 움직임 성분의 성격을 파악할 수 있는 방법에 대하여 살펴본다.

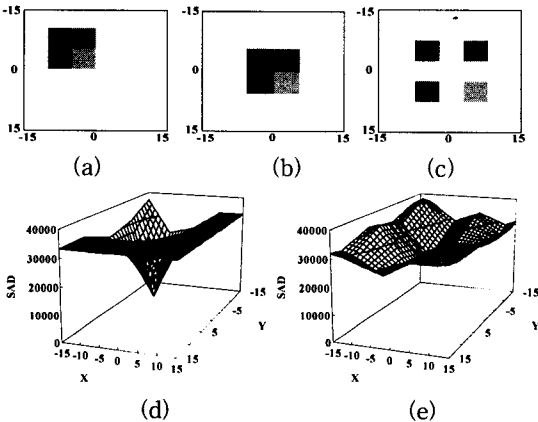


그림 5. 움직임 추정 오차 분포 (a) 선형 움직임 탐색 영역 (b) 현재 화면의 매크로 블록 (c) 비선형 움직임 탐색 영역 (d) 탐색 영역(a)의 EDME (e) 탐색 영역(c)의 EDME

Fig. 5. Error distribution of motion estimation (EDME). (a) searching area for linear motion (b) macroblock of current frame (c) searching area for nonlinear motion (d) EDME of searching area (a) (e) EDME of searching area (c)

2) 움직임 판별

매크로 블록 내에 존재하는 움직임 성분의 성격을 파악하기 위하여 앞에서 설명된 EDME를 이용하였다. EDME의 분포 형태를 분류하기 위하여 그림 6과 같이 탐색 영역을 네 개의 영역  $PART_i(i=1,2,3,4)$ 로 구분하여 각 영역의 최소  $SAD_i(i=1,2,3,4)$ 와 매크로 블록 단위로 추정된 움직임 벡터 위치의  $SAD_{min}$ 값의 비를 이용하였다.  $SAD_i/SAD_{min}$ 은 그림 5(d)와 같은 EDME의 경우 3개 영역에서 큰 값을 가지며, 그림 5(e)와 같은 EDME에서는 4개의 값이 모두 작은 값을 가지게 된다. 이는 그림 5(c)의 탐색 영역에서 매크로 블록 단위의 움직임 추정이 예측 오차를 크게 함을 나타내며, 블록단위의 움직임 추정이 필요함을 의미한다. 따라서, 본 논문은 이러한  $SAD_i/SAD_{min}$ 의 값이 작은 영역의 개수가 3개 이상이라면 블록 단위의 움직임 추정을 하고 그렇지 않은 경우에 이러한 과정을 생략할 수 있도록 하여 부호화 효율을 향상시켰다. 식 3은 탐색 영역의 각 분할 영역의 최소값  $SAD_i$ 를 구하기 위한 과정이며, 식 4는 구해진  $SAD_i$ 와  $SAD_{min}$ 의 비가 임계값  $a$ 보다 작은 영역의 개수  $SP$ 를 구하는 식이다. 여기서,  $a$ 는 각 분할 영역의  $SAD_i$ 값들의 유사성을 판별하는 기준이 되며, 이 값이 큰 값을 가지면, APM 모드의 선택 가능성이 높아지게 되며, 작은 경우에는 DPM의 선택 가능성을 높게 한다. 임계값  $a$ 에 대한 결정은 다음절에 논의한다.

$$SAD_i = \min(SAD(x)) \quad x \in PART_i \quad (3)$$

$$cp_i = \begin{cases} 1 & \text{if } \frac{SAD_i}{SAD_{min}} < a \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$SP = \sum_i cp_i$$

그림 6의 각 영역간에 빗금친 부분의 간격을 둔 것은 각 영역의 경계 부분에 추정된 움직임 벡터가 존재하는 경우에는 유사한 값이 두 개 이상의 영역에서 구해지기 때문이다. 이러한 간격이 없는 경우에 만일 (0,0)에서  $SAD_{min}$ 이 결정된다면, (0,0) 주변의 값이 이와 유사한 값이 되어 각 영역의 최소값  $SAD_i$ 가  $SAD_{min}$ 의 값과 유사한 값을 가지게 된다. 이와 같이 최소값이 여러 영역에 나타나는 것을 막기 위한 것이다. 또한 매크로 블록의 영상이 변화가 거의 없는 텍스처 성분이 탐색 영역 전반에 걸쳐 나타나는 경우

에도 EDME가 그림 5(e)와 유사한 분포를 보이는데 이러한 경우에는 예측 오차 값이 작게 된다. 따라서, 예측 오차 값이 작은 경우에도 블록 단위의 움직임 추정 은 이루어지지 않도록 하였다.

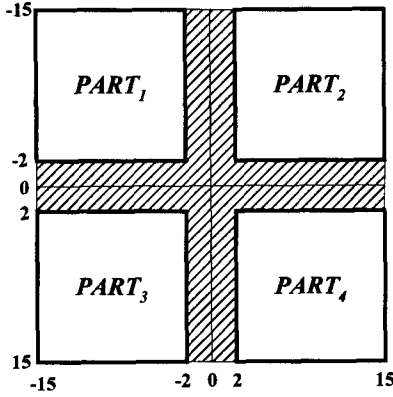


그림 6. 움직임 추정 오차 분포 영역의 분할  
Fig. 6. Partition of region in error distribution of motion estimation.

3) 임계값  $\alpha$  의 결정

그림 7은  $\alpha$  값의 변화에 따른 DPM에 대한 화질의 개선정도를 나타낸 것이다. 그림 7에서 살펴볼 수 있듯이  $\alpha$  값이 커질수록 화질의 개선이 두드러지고, 일정값 이상에서 더 이상 화질의 개선이 이루어지지 않게 된다. 이는 대부분의 화면이 APM으로 부호화 됨을 의미한다. 주어진 전송률이 높아질수록 화질의 개선이 더 이상 이루어지지 않는  $\alpha$  가 큰 값으로 나타나는 것을 살펴볼 수 있다.

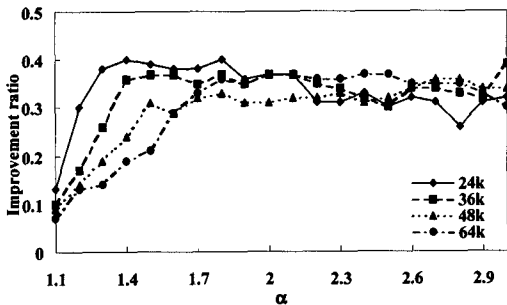


그림 7.  $\alpha$  값에 의한 DPM과의 PSNR비(Suzie 150frames)  
Fig. 7. Comparison of PSNR with DPM according to  $\alpha$ .

본 논문은 이러한 주어진 비트율에 따라  $\alpha$  값이 조정될 수 있도록 식 5를 사용하였다.

$$\alpha = 0.2 \left( \frac{T_b}{12000} \right) + 1 \tag{5}$$

여기서,  $\alpha$  는 각 분할 영역의 SAD 값들의 유사성을 결정하는 임계값이며,  $T_b$  는 목표 전송률이다. 결과적으로  $\alpha$  값이 1.2 이하와 2.1 이상에서 화질의 개선이 두드러지지 않기 때문에,  $\alpha$  값은 이 사이에서 선택되는 것이 바람직할 것이다. 식 5는  $T_b$  의 값이 각각 24k, 36k, 48k, 64k로 주어질 때  $\alpha$  는 각각 1.4, 1.6, 1.8, 2.07의 값이 선택된다.

2. 움직임 추정

그림 8(a)는 H.263부호화기의 APM으로 부호화 되는 경우의 움직임 추정 방법의 순서도를 나타내었다. 그림에서 보여지듯이 매크로 블록 단위의 움직임 추정과 블록 단위의 움직임을 수행하여 두 방법의 예측 오차  $SAD_{16}$  과  $SAD_8$  을 비교한 후,  $SAD_{16}$  이 작다면 하나의 움직임 벡터만을 생성하고,  $SAD_8$  이 작은 경우는 블록 단위로 네 개의 움직임 벡터를 생성한다. 기존의 방법은 두 가지 추정 방법을 모두 수행하기 때문에 움직임 추정시 DPM에 비하여 1.25배 정도의 계산량을 필요로 한다. 이는 매크로 블록 단위의 움직임 추정은  $\pm 15$ 의 탐색영역에 대하여 수행되며, 블록 단위의 움직임 추정은  $\pm 7$ 의 탐색영역에 대하여 수행되기 때문이다. 반면에, 그림 8(b)의 제안한 움직임 추정 방법은 우선 매크로 블록 단위의 움직임 추정시 형성

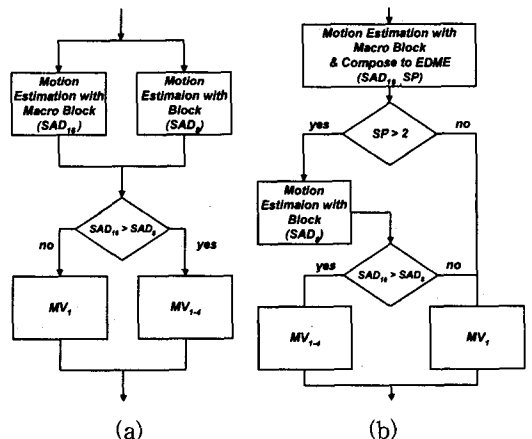


그림 8. 움직임 추정 방법의 비교  
(a) APM을 사용시 H.263의 움직임 추정 방법 (b) 제안한 움직임 추정 방법  
Fig. 8. Comparison of Motion Estimation Method. (a) Motion Estimation in H.263 using APM (b) Motion Estimation in proposed coding scheme

되는 EDME를 분석하여 매크로 블록내에 존재하는 움직임을 반영하는 SF의 값에 따라 블록 단위의 움직임 추정 여부를 결정하여 계산량을 줄일 수 있었다.

### 3. APM의 동적 선택

APM으로 부호화 되는 경우에는 중복 움직임 보상에 의하여 계산량이 증가하게 된다. 또한 선형적인 움직임만이 화면 내에 존재하는 경우에 대하여 중복 움직임 보상 방법은 DPM으로 부호화 하는 경우에 비하여 부호화 효율을 떨어뜨릴 수 있다. 본 논문에서 제안한 부호화기는 움직임 추정시 블록단위의 움직임 추정의 필요성에 따라 APM과 DPM이 동적으로 선택된다. 전체 매크로 블록에 대하여 블록 단위의 움직임 추정이 이루어지지 않았다면 다음 화면을 DPM으로 부호화 하여 계산량을 줄이고, 부호화 효율이 높아지도록 하였다. DPM으로 부호화 되는 과정에서 EDME의 형태를 분석하여 블록 단위의 움직임 추정이 필요한 매크로 블록이 존재한다면, 다음 화면을 APM으로 부호화 한다. 즉, DPM으로 부호화 되는 과정에 SF의 값을 구하여 블록단위의 움직임 추정의 필요성을 파악하고, 블록 단위의 움직임 추정은 수행되지 않는다. 이 값을 계산하기 위한 계산량은 매크로 블록당 4회의 나눗셈과 4회의 비교로 추가되는 계산량은 미미하다 할 것이다. 결과적으로 화면 내에 존재하는 움직임 성분을 EDME의 분포에 따라 SF의 값을 구하여 다음 화면을 APM과 DPM을 선택적으로 적용하여 부호화 한다. 따라서, APM으로 부호화 할 필요가 없는 화면을 DPM으로 부호화 하여, APM의 부호화 과정에서 발생하는 부호화 효율의 저하를 막으며 추가되는 계산량을 줄일 수 있으며, 화질면에서는 APM을 전체적으로 적용한 것과 같은 결과를 얻을 수 있다.

## IV. 실험 결과

제안한 방법의 부호화 효율을 실험하기 위하여, 주어진 목표 비트율에 대하여 PSNR 관점에서 몇 가지 비디오 열에 대한 결과를 그림 9에 나타내었다. 실험 영상은 QCIF 형식의 영상이고 목표 비트율은 24kbps이다. 또한 Skip 화면 수는 2로하여 초당 10화면을 부호화 하도록 하였다. 동일한 실험 영상에 대하여, DPM, 고정된 APM(Fixed APM, FAPM)으로 부호

화 된 결과와 제안한 동적 APM(Dynamic APM, DAPM)을 화면별로 PSNR의 변화를 나타내었다.

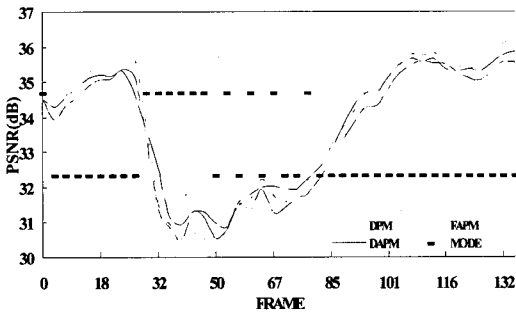
그림 9(a), (b), (c)는 각각 실험 영상 Suzie, Carphone, Coast 150화면에 대한 결과이다. 결과적으로, 제안한 DAPM은 대부분의 화면에 있어 FAPM으로 부호화 한 것과 비슷한 결과를 얻을 수 있었다. 그림의 MODE는 위 부분으로 올라간 경우가 APM이 선택된 부분이고, 반대의 부분이 DPM이 적용된 부분이다. 그림 9(a)의 결과에서 보여지듯이 화면 내에 비선형적인 움직임이 많이 발생하는 32~70화면에 대하여 APM이 동적으로 선택되어 고정된 모드로 부호화 된 결과에 비해 화질을 향상시키고 있음을 확인할 수 있다.

표1에서 제안한 DAPM의 성능을 PSNR과 초당 부호화된 화면 수(Frames Per Second, FPS)의 관점에서 각각 DPM과 FAPM으로 부호화한 결과와 비교하였다. 결과적으로 PSNR은 FAPM으로 부호화한 경우와 비슷한 결과를 얻을 수 있었다. 또한 FPS는 Suzie영상을 64kbps로 부호화한 결과를 제외하고는 전체적으로 제안한 방법이 FAPM의 결과에 비해 우수함을 나타내었다. 각각의 실험 결과는 Suzie영상 150화면, Carphone영상 382화면, Coast영상 300화면을 10Hz로 부호화한 결과이다. 제안한 APM을 동적으로 선택하여 부호화 하는 DAPM방법은 저 전송률에서 보다 더 유용하게 적용될 수 있음을 보여 준다.

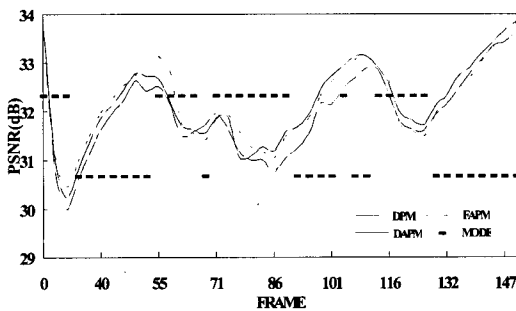
표 2는 FAPM과 DPM에 대한 움직임 추정시 전역 탐색을 사용하였을 경우 탐색점의 수를 백분율로 나타내었다. FAPM과 DPM으로 부호화 하는 경우의 움직임 추정시 탐색점의 수를 각각 100%로 하였을 경우에 대하여 제안한 방법의 탐색점의 수를 백분율로 표시한 것이다. 표 2의 결과는 제안한 방법이 APM으로 부호화 했을 경우에 비해 83~89% 정도의 탐색점에 대한 계산으로 움직임 추정 과정이 진행됨을 나타낸다. 이는 매크로 블록단위로 추정된 움직임 오차를 바탕으로 블록단위의 움직임 추정 여부를 판단하기 때문에 얻어진 이득이다. 반면에 DPM으로 부호화 했을 경우에 비해 추가되는 탐색점의 수는 3~10%정도임을 나타낸다.

중복 움직임 보상이 사용되는 경우, 화소당 4번의 곱셈과 2번의 덧셈이 필요하다. 즉 FAPM으로 부호화 하는 경우, QCIF 화면에 대하여, 한 화면을 부호화 하는데 있어 101,376(176×144×4)회의 곱셈과

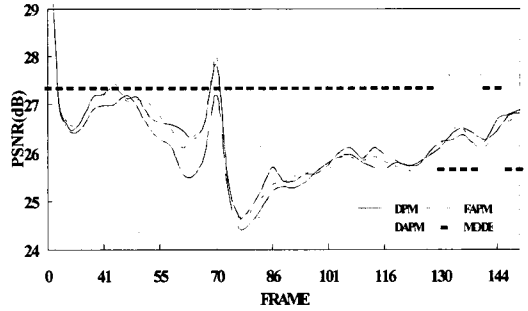
50,688(176×144×2)회의 덧셈이 추가적으로 필요하게 된다. 이 양을 100%로 했을 경우에 DAPM의 경우 추가되는 계산량을 표 3에 백분율로 나타내었다. 이 결과는 전체 부호화된 화면 중에 APM으로 부호화된 화면의 비율과 같다. 즉, 전체화면 중에 APM으로 부호화 되는 경우가 30%를 나타낸다면, 나머지 70%의 화면은 DPM으로 부호화 되기 때문에 추가적으로 발생하는 계산을 30%정도로 줄일 수 있음을 나타낸다. 예를 들어, Suzie영상의 경우 총 42개의 부호화 된 화면 중 11개 화면이 APM으로 부호화 되고, 나머지 31화면은 DPM으로 부호화 되기 때문에 11개 화면만이 추가적인 계산이 필요하게 된다. 이는 FAPM에 비해 26.19%의 계산량으로 부호화 할 수 있음을 나타낸다. 제안한 DAPM은 부호화 할 영상의 특징에 따라 APM과 DPM이 동적으로 적용되어 부호화 된다. 결과적으로, FAPM으로 부호화 하는 경우에 발생하는 추가적인 계산을 줄일 수 있었으며, 화질 면에서 FAPM으로 부호화 한 결과와 비슷한 화질을 얻을 수 있었다. 또한 부분적으로 화질이 떨어지는 부분에서는 FAPM보다 나은 화질을 얻을 수 있었다.



(a) Suzie (150 frames)



(b) Carphone (150 frames)



(c) Coast (150 frames)

그림 9. PSNR의 비교

Fig. 9. Comparison of performance.

표 1. 평균 PSNR과 초당 부호화 화면수의 비교

Table 1. Comparison of mean PSNR and FPS.

sequence	frames	bitrate	DPM		FAPM		DAPM	
			PSNR (dB)	FPS (Hz)	PSNR (dB)	FPS (Hz)	PSNR (dB)	FPS (Hz)
Suzie	150	24kbps	33.53	9.02	33.79	9.05	33.78	9.11
		36kbps	35.03	9.06	35.39	9.15	35.29	9.15
		48kbps	36.12	9.17	36.48	9.17	36.39	9.17
		64kbps	37.16	9.25	37.60	9.31	37.53	9.25
Carphone	382	24kbps	29.80	9.80	30.02	9.66	29.94	9.83
		36kbps	31.32	10.00	31.52	9.73	31.47	10.03
		48kbps	32.38	9.97	32.62	9.97	32.52	9.97
Coast	300	24kbps	26.64	10.15	26.78	10.05	26.77	10.25
		36kbps	27.97	10.39	28.25	10.18	28.19	10.39
		48kbps	28.94	10.38	29.27	10.14	29.22	10.34
		64kbps	29.95	10.38	30.30	10.17	30.24	10.31

표 2. DAPM의 탐색점의 수(%)

Table 2. The number of searching point in DAPM(%).

sequence	화면수	비트율	FAPM에 대한 탐색점수(%)	DAPM에 대한 탐색점수(%)
Suzie	150	24kbps	82.77	103.46
		36kbps	83.27	104.08
		48kbps	83.73	104.66
		64kbps	84.29	105.37
Carphon c	382	24kbps	82.92	103.64
		36kbps	83.57	104.46
		48kbps	84.08	105.10
		64kbps	84.70	105.88
Coast	300	24kbps	86.53	108.17
		36kbps	87.11	108.89
		48kbps	87.77	109.72
		64kbps	88.77	110.96



표 3. DAPM의 움직임 보상시 계산량  
Table 3. Computational complexity of motion compensation in DAPM.

sequence	화면수	비트율	부호화된 화면수	APM으로 부호화된 화면수	FAPM에 대한 계산량
Suzie	150	24kbps	42	11	26.19 %
		36kbps	44	9	20.45 %
		48kbps	45	7	15.56 %
		64kbps	46	8	17.39 %
Carphone	382	24kbps	118	42	35.59 %
		36kbps	123	36	29.27 %
		48kbps	124	37	29.84 %
		64kbps	125	34	27.20 %
Coast	300	24kbps	95	65	68.42 %
		36kbps	99	49	49.49 %
		48kbps	100	34	34.00 %
		64kbps	101	25	24.75 %

### V. 결론 및 고찰

본 논문에서는 H.263 부호화기의 선택 사양으로 되어있는 APM을 동적으로 선택할 수 있는 방법을 제안하였다. 이를 위하여 움직임 추정 오차 분포(EDME)를 구성하여, 이 분포로부터 매크로 블록의 움직임 성분의 성격을 파악할 수 있는 방법을 제시하였으며, 이를 바탕으로 블록 단위의 움직임 추정 과정을 생략하거나, DPM으로 부호화 하도록 하여 중복 움직임 보상시 발생하는 추가적인 계산을 줄일 수 있도록 하여 부호화 효율을 향상시켰다. 제안한 방법은 영상의 특성에 따라 H.263의 APM이 동적으로 선택되어 부호화 되며, 화질이 저하되거나, 영상 내에 비선형적인 움직임이 존재하는 부분에 APM이 동적으로 선택되어 부호화 됨을 확인할 수 있었다. 특히 저전송율에서 보다 효율적으로 동작하는 것을 확인할 수 있

었으며, 높은 전송률이나 움직임이 전반적으로 적은 영상의 경우 APM모드의 선택이 상대적으로 적게 되는데 이는 화질이 일정 정도 유지된다면, 부호화기를 단순하게 동작시켜도 된다는 사실을 반영한 것이다. 제안한 동적 모드 부호화기는 영상 특성의 변화가 심한 실제적인 비디오 열에 보다 효율적일 것으로 생각되며, H.263의 다른 선택 사항에 대해서도 동적으로 선택할 수 있는 부호화기의 설계가 가능할 것이다.

### 참고 문헌

- [1] H.Gharavi and M.Mills, "Block Matching Motion Estimation Algorithm - New Results," IEEE Trans., Circuits and Systems, vol. 37, No.5, pp 649-651, May 1990.
- [2] D. Le Gal, "MPEG : A Video Compression Standard for Multimedia Application," Commun. of ACM, vol. 34, pp. 47-58, Apr. 1991.
- [3] Draft ITU-T Recommendation H.261, "Video Codec for Audiovisual Services at p ×64kbit/s.," Aug. 1990.
- [4] Draft ITU-T Recommendation H.263, "Video Coding for Low Bitrate Communication," 2 May, 1996.
- [5] Tae Won Hur, Duk Chul Moon, Keun Young Lee, "A Video Sequence Coding Using Dynamic Selection Of Unrestricted Motion Vector Mode In H.263 Bitstream," Proceedings MoMuc '97. pp. 352-355, Oct. 1997.

### 저자 소개



許泰元(正會員)  
1967年 7月 8日 出生 1990年 2月  
成均館大學校 電子工學 工學士  
1993年 2月 成均館大學校 電子工學  
工學碩士 1996年 2月 成均館大學  
校 電子工學 博士修了. 주 관심분야  
는 영상 부호화, 신호처리, 디지털시스템설계

李根泳(正會員) 第 35 卷 S編 第 3 號 參照  
현재 성균관 대학교 전기전자 컴  
퓨터 공학부장