

論文94-31B-8-14

# 탐색 영역의 적응적 이동에 관한 연구

## (A Study on Adaptive Moving Method of Search Region)

金珍台\*, 李錫鎬\*\*, 崔宗秀\*

(Jin Tae Kim, Seok Ho Lee, and Jong Soo Choi)

### 要約

본 논문에서는 움직임에 따라 탐색 영역을 적응적으로 이동시키는 방법을 제안한다. BMA에서 탐색 영역은 하드웨어 구현 가능성과 움직임의 정도에 따라 결정되는데 한번 결정되면 부호화하는 도중에 변경시킬 수 없다. 본 논문에서는 부가 정보 없이 이전 프레임의 움직임 벡터를 이용하여 현재 블록의 움직임 정도를 예측하고, 예측된 정보를 바탕으로 움직임 정도에 따라 탐색 영역의 위치를 이동시킨다. 따라서 제안한 방식은 움직임이 크면 탐색 영역의 크기를 크게 할 수 있는 효과를 가질 수 있게 된다. 실험 결과, 움직임이 큰 블록에서 기존의 BMA보다 예측 오차를 감소시켜 예측 효율을 개선하였다.

### Abstract

In this paper, an adaptive moving method of the search region tracking the motion is proposed. The search region in BMA is determined by the capability of hardware implementation and the degree of motion. But once determined, nothing can be changed during coding procedure. In this paper, we predict the level of motion of the current block using motion vectors of previous frames without overhead information, and change the location of the search region according to the level of motion predicted. In short, the proposed method can be achieved the desirable effect such that the size of search region gets large when the motion is large. Results of experiments show that prediction efficiency has been improved by using adaptive moving method resulting in reduced prediction error in the blocks with large motion.

### 1. 서론

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

\*\*正會員, 韓國通信 소프트웨어 研究所  
(Software Research Lab., Korea Telecom.)

接受日字 : 1993年 9月 1日

영상 부호화의 목적은 가능한한 최소의 비트로 원 영상을 충실히 표현하고 재생하는데 있다. 영상 신호에는 이웃 화소간의 상관성(correlation)으로 인하여 중복성(redundancy)이 존재하는데, 이러한 중복성

을 제거함으로써 영상 데이터의 압축이 가능하게 된다.<sup>1)</sup>

시간에 따라 변하는 동영상(moving image)은 공간상의 상관성과 시간상의 상관성을 가지고 있어 효율적으로 많은 양의 데이터를 압축할 수 있다. 동영상의 경우 시간상의 상관 계수가 공간상의 상관 계수보다 훨씬 크기 때문에 시간상의 상관성을 이용하는 프레임간(interframe) 부호화가 효율적이다. 프레임간 부호화 방법은 움직임 추정(motion estimation)에 의한 움직임 보상 부분과 예측 오차(prediction error)를 부호화하는 부분으로 구성된다.<sup>2)</sup> 대부분의 시스템<sup>3,4)</sup>의 움직임 추정부에 사용되는 블럭 정합 알고리즘(block matching algorithm: BMA)은 이전 프레임의 탐색 영역(search region)에서 현재 프레임의 블럭과 평가 함수가 최적인 위치를 움직임 벡터로 결정하는 방식이다.

본 논문에서는 이전 프레임의 탐색 영역을 움직임에 따라 적응적으로 이동시켜 예측 오차의 감소를 목적으로 한다. BMA에서 탐색 영역의 크기가 결정되면 부호화하는 도중에는 크기를 변화시킬 수 없으므로 탐색 영역을 벗어난 움직임이 있는 경우의 움직임 추정은 부정확하다. 본 논문에서는 이전 프레임들의 움직임 벡터를 사용하여 현재 프레임 블럭의 움직임을 예측하고, 예측된 결과를 바탕으로 탐색 영역을 이동시킨다. 탐색 영역의 이동은 이전 두 프레임의 움직임 벡터의 크기와 방향을 이용하여 예측을 하므로 복호기에서 같은 처리를 할 수 있다. 그러므로 부가 정보(side information)가 필요치 않다. 제한한 방법은 움직임이 존재하는 블럭들에 대해 같은 크기의 탐색 영역을 사용한 기존의 움직임 추정 방법보다 예측 오차를 감소시킨다.

본 논문의 구성은 I 장 서론에 이어, II 장에서는 telescopic 탐색에 대해 기술하며, III 장에서는 제안한 방법에 대한 설명을 한다. 그리고 IV 장에서는 컴퓨터 시뮬레이션과 그 결과를 검토하며, V 장에서 결론을 맺는다.

## II. Telescopic 탐색

Telescopic 탐색<sup>5)</sup>은 MPEG에서 제안된 방식으로, 프레임사이의 간격이 클때 탐색영역이 커져 증가되는 계산량을 막기 위해 이전에 구해진 움직임 벡터를 사용하여 현재의 움직임을 추정하는 방식이다. 인접한 두 프레임에서 탐색 영역의 최대 이동 변위를  $\pm 15$ 로 가정하고, 연속되는 프레임이 IBBP 구조를 갖는다고 하자(2개의 보간 프레임이 있는 경우). 이

경우에 부호화는, I 프레임, P 프레임 그리고 2개의 B 프레임 순서로 부호화된다. 이때 움직임을 추정하는 프레임 간격이 2 또는 3이 되는 경우의 최대 이동 변위는 2 프레임 간격이면  $\pm 30$ 이 되고 3 프레임 간격이면  $\pm 45$ 가 되어 탐색 영역이 커지고 계산량이 증가하게 된다. 계산량을 감소시키기 위하여 이전에 구해진 움직임 벡터를 사용하여 탐색 영역의 크기를 한 프레임 간격인  $\pm 15$ 로 줄여 움직임을 추정하는 방식이 telescopic 탐색이다. 그림 1은 telescopic 탐색과 전역 탐색(full search)을 비교한 그림이다. IPPP 구조를 갖는 형태에서는 움직임을 추정하는 프레임 간격이 1이기 때문에 telescopic 탐색을 적용시킬 필요는 없으나 이전 프레임의 움직임을 효율적으로 현재 프레임에 이용한다는 측면에서 telescopic 탐색의 개념을 도입할 수 있다. 즉 이전 프레임의 움직임 정보를 현재 프레임의 움직임 추정시 탐색 영역의 이동에 이용한다.

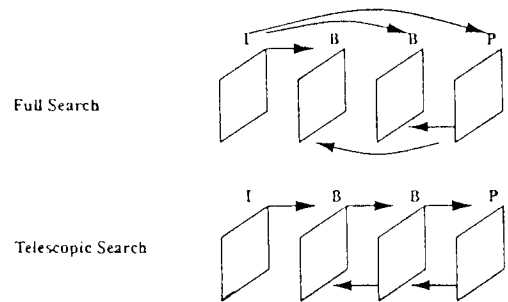


그림 1. Telescopic 탐색과 전역 탐색<sup>5)</sup>  
Fig. 1. Telescopic search and full search.<sup>5)</sup>

Telescopic 탐색이 이전 프레임의 움직임 벡터를 offset 벡터로 이용하는 것에 비해 제한한 방법은 이전 두 프레임의 움직임 벡터를 이용하여 현재 프레임 블럭의 움직임의 정도를 계산하여 이를 탐색 영역 이동에 사용한다.

## III. 적응 탐색 영역 추정

제한한 방법에서 탐색 영역의 이동은 이전 프레임의 움직임 벡터를 이용한다. 이때 현재 블럭의 속도를 추정하기 위해 이전 두 프레임의 움직임 벡터가 필요하다. 그림 2에서 처럼 현재 프레임  $F_t$ 의 탐색 영역은 이전 프레임  $F_{t-1}$ 과  $F_{t-2}$  사이의 움직임 벡터  $V_{t-1}$ 과 이전 프레임  $F_{t-2}$ 과  $F_{t-3}$  사이의 움직임 벡터  $V_{t-2}$ 를 이용한다.

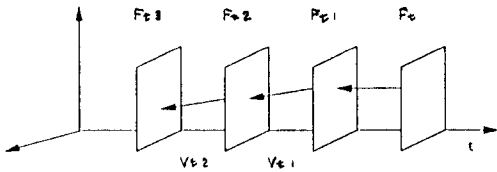


그림 2. 프레임의 순서  
Fig. 2. Sequence of frame.

1. 움직임 블럭과 배경 블럭의 구분

BMA에 의한 움직임 벡터는 단지 최소의 오차를 갖는 위치를 결정하므로 배경 영역에서도 큰 움직임 벡터를 갖는 블럭이 존재한다. 그러나 배경 영역은 조명의 변화와 카메라의 잠움 등에 의하여 변화할뿐 움직임이 없기 때문에 탐색 영역의 이동은 필요없는 예측 오차의 증가를 가져온다. 그래서 배경 블럭과 움직임 블럭의 구별이 필요하게 된다. 배경 블럭의 결정은 이전 프레임에서 같은 위치 블럭의 움직임 벡터와 주변의 8개 블럭의 움직임 벡터를 조사하여 결정한다. 즉, 이전 프레임의 블럭에 대해 식 (1)과 같이 9개 블럭의 움직임 벡터의 평균을 구한다.

$$M = \frac{1}{9} \sum_{m=-1}^1 \sum_{n=-1}^1 |MV_{t-1}(i-m, j-n)| \quad (1)$$

여기서,  $MV_{t-1}(i, j)$ 는 t-1 프레임의 i, j 위치 블럭의 움직임 벡터이다. 식 (1)에서 구해진 값을 식 (2)와 같이 문턱치와 비교하여 배경 블럭과 움직임 블럭을 구분한다.

$$\begin{cases} M \geq T & : \text{움직임이 있는 블럭} \\ M < T & : \text{배경 영역의 블럭} \end{cases} \quad (2)$$

그러나 배경 영역에서 움직임 블럭이 단독으로 존재하는 경우가 발생할 수 있으므로 식 (3)과 같이 이전 프레임에서 같은 위치에 있는 블럭의 움직임 벡터에 같은 문턱치를 적용하여 배경 영역에 존재하는 움직임이 있는 블럭을 고려한다. 문턱치 보다 크면 움직임이 있는 블럭으로 간주한다.

$$|MV_{t-1}(i, j)| \geq T : \text{움직임 블럭} \quad (3)$$

움직임이 있다고 판단된 블럭은 다음의 과정에 의하여 탐색 영역의 추정이 이루어지고, 이동된 탐색 영역을 기준으로 움직임 추정이 행해진다. 배경 블럭은 탐색 영역의 이동없이 현재 위치를 기준으로 하여 움직임 추정이 행해진다. 즉, 배경이라고 판단된 블

럭은 기존의 BMA와 같은 탐색 영역에서 움직임 추정을 한다.

2. 탐색 영역의 이동

BMA는 모양의 변화가 없는 물체의 병진 운동만이 존재한다는 가정하에 이루어진다. 그러나 실제의 영상에서는 여러가지 형태의 움직임이 나타나고 때로는 급격하고 빠른 움직임이 존재한다. 이런 경우에 BMA의 효율이 감소하기 때문에 이전 프레임의 움직임 정도에 따라 현재 프레임의 움직임 추정시 탐색 영역을 이동시킨다.

움직이는 물체는 가속도에 의하여 그 다음의 움직임을 어느정도 예측할 수 있다. 운동의 법칙에 의하면, 움직임이 불규칙하더라도 움직임의 방향이 변하기 위하여는 속도가 0(zero)이 된후에 다른 방향으로의 가속이 이루어진다. 일반적인 영상의 경우, 프레임사이의 시간 간격이 짧기(초당 30 ~ 60 프레임) 때문에 속도의 변화를 감지할 수 있고, 물체의 움직임이 있으면 그 블럭의 속도를 추정할 수 있다. 속도를 계산하기 위해 2개의 움직임 벡터를 참조한다.

i) 블럭의 속도  $\bar{v}$ 는 식 (4)와 식 (5)처럼 구한다.

$$\bar{v}_{t-2} = \frac{l_{t-2} - l_{t-3}}{\Delta t} = \frac{MV_{t-2}}{\Delta t} \quad (4)$$

$$\bar{v}_{t-1} = \frac{l_{t-1} - l_{t-2}}{\Delta t} = \frac{MV_{t-1}}{\Delta t} \quad (5)$$

$\Delta t$  : 프레임 사이의 시간 간격

$l_{t-1} - l_{t-2}$  : 프레임 사이의 움직임 벡터 거리

$MV_{t-1}, MV_{t-2}$  : 움직임 벡터

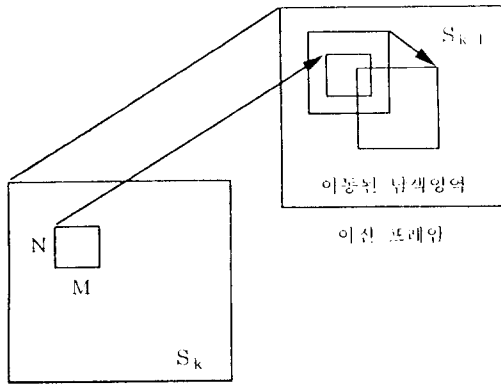
ii) 속도의 비율을 계산하면 다음과 같다.

$$dv = \left| \frac{\bar{v}_{t-1}}{\bar{v}_{t-2}} \right| \quad (6)$$

iii) 움직임의 추정은 움직임 벡터의 크기에 비례하다고 할 수 있으므로 식 (6)의 dv에 비례하여 다음과 같이 구한다.

$$S = dv \cdot MV_{t-1} \quad (7)$$

위와 같은 과정을 수평(x축), 수직(y축) 방향에 대하여 실행하여 현재 블럭의 위치에서 정해진 탐색 영역을 식 (7)에서 구해진  $S_x, S_y$ 만큼 x, y축으로 이동시킨다. 그림 3은 이동된 탐색 영역을 나타낸다. 이렇게 정해진 탐색 영역을 바탕으로 움직임 추정이 행하여진다.



현재 프레임

그림 3. 이동된 탐색 영역

Fig. 3. Moved search region.

3. 보정

일반적인 영상에서는 물체의 모양 변화가 심하고 병진 운동과 함께 회전 운동도 나타나게 된다. 그리고 BMA로 구한 움직임 벡터는 신뢰성의 문제가 있어 블록의 속도를 계산하는데 오차 발생의 원인이 된다. 다음의 두 경우에 움직임 검출에 오차가 발생했다고 볼 수 있다. 그래서 움직임 벡터에 의한 탐색 영역의 이동은 위와 같은 경우에 보정이 필요하게 된다.

경우 1)  $MV_{t-1}$  와  $MV_{t-2}$ 의 운동 방향이 반대이고 두 값의 차가 큰 값을 갖는 경우

경우 2)  $MV_{t-1}$  와  $MV_{t-2}$ 의 방향은 같으나 두 값의 비가 큰 값을 갖는 경우

1) 운동 방향이 반대인 경우

탐색 영역을 이동시키는 방법에서 움직이는 블록의 속도가 가정에 의하면 움직임의 방향이 변할때는 움직임 속도가 0 정도가 되고 다시 반대의 방향으로 움직인다. 움직이는 방향이 변할때, 그 부분의 이동 정도가 크다면 움직임 검출에 오차가 생겼다고 볼 수 있다. 그래서 이 부분을 보정하기 위해 다음 과정이 행해진다.

현재 블록의 움직임 벡터의 방향과 크기는 그 블록의 주위 8 블록의 움직임 벡터를 참조하여 결정한다. 움직임 벡터 평면에서  $3 \times 3$  마스크(mask)를 사용하여 방향의 우위를 결정한다. 방향의 우위 결정은 + 방향인지 - 방향인지를 검사한다. 우위 방향이 결정되면 그 방향을 현재 블록의 움직임 방향으로 설정한다. 현재 블록의 움직임 벡터 크기는 우위 방향의 움직임 벡터들로부터 식 (8)과 같이 구한다.

$$K = \frac{1}{N} \sum_{i,j} MV_{t-1}(u,v) \quad (8)$$

여기서  $MV_{t-1}(u,v)$ 는 우위 방향에 속하는  $u,v$  위치 블록의 움직임 벡터이다.  $K$ 는 우위 방향 블록들의 움직임 벡터 평균값을 의미한다. 따라서 현재 위치의 움직임 벡터를  $K$ 로 대체하고 이를  $MV_{t-1}^*$ 라 한다.

$$MV_{t-1}^*(i,j) = K \quad (9)$$

이렇게 정해진  $MV_{t-1}^*$ 는 탐색 영역의 이동을 결정하기 위해서 식 (4)와 식 (5)에서 쓰인다.

2) 운동 방향이 같은 경우

탐색 영역의 이동의 요인은 움직임 벡터  $MV_{t-2}$ 와  $MV_{t-1}$ 에 의하여 결정된다. 그러나 움직임 벡터  $MV_{t-2}$ 와  $MV_{t-1}$ 의 비가 클때는 가정에 모순이 되고 이 블록의 탐색 영역의 이동이 상당히 크게 된다. 이것은 움직임 벡터의 오차에서 기인한다고 볼 수 있다. 즉, BMA에 의한 움직임 추정시 단순히 최소의 예측 오차를 갖는 위치를 찾아내므로 실제의 움직임과 차이를 나타내는 경우이다. 그래서 탐색 영역의 이동 거리를 감소시키고 오차를 줄이기 위해서 다음의 함수를 사용한다.

$$S = \sqrt{dv \cdot \bar{v}_{t-1}} \quad (10)$$

여기서,  $S$ 는 탐색 영역의 이동량이고,  $\bar{v}_{t-1}$ 는 블록의 속도이다. 그림 4는 탐색 영역 이동 방법의 순서도이다.

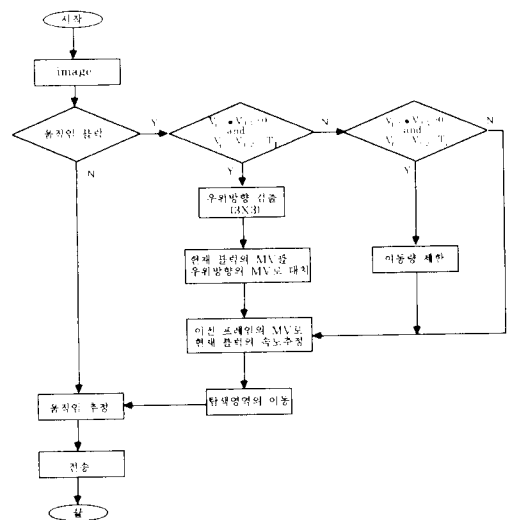


그림 4. 탐색 영역 이동 방법의 흐름도

Fig. 4. Flowchart of the adaptive moving method of search region.

IV. 컴퓨터 시뮬레이션 결과 및 검토

본 실험에서는 720×480 크기의 MPEG 표준 영상인 'Football' 과 'Flower Garden' 영상을 사용하였다. 실험에 사용한 영상은 CCIR 601 format의 4:2:2의 디지털 신호로 구성되어 있으며, 원래의 영상은 비월주사된 형태이나 본 실험에서는 MPEG의 해상도 변화에 사용하는 보간 필터<sup>3)</sup>로 순차주사의 형태로 바꾸어 휘도 신호만을 대상으로 하였다.

적응적으로 탐색 영역을 결정하기 위해서는 이전 두 프레임의 움직임 벡터가 필요하다. 따라서 적응적으로 탐색 영역이 이동되는 경우는 4번째 프레임부터이다. 제안한 방법은 부호기의 탐색 영역 이동에 사용되는 움직임 벡터들은 복호기에서도 이미 저장되어 있으므로 부호기와 똑같은 작업을 할 수 있다. 즉, 복호기에서도 부호기와 똑같이 탐색 영역을 이동시킬 수 있기 때문에 탐색 영역의 이동에 따른 움직임 벡터의 전송량의 증가나 부가 정보가 필요치 않다.

실험 결과로는 제안한 방법과 telescopic 탐색, 전역 탐색 등을 비교하였다. 본 실험에서는 영상이 IPPP 구조라고 가정하였다. 이 경우에 전역탐색의 탐색영역 크기는 매 프레임에서 ±15가 되고, telescopic 탐색에서 탐색영역의 이동량은 이전 프레임의 움직임 벡터를 1/2하여 이를 offset 벡터로 하였다. 탐색영역의 크기는 전역탐색과 같다. 탐색 영역을 이동시킬때 사용한 움직임 벡터는 2가지 형태를 이용하였다. 원래의 BMA로 구한 움직임 벡터들을 다음 프레임의 탐색 영역의 이동에 이용한 경우와 움직임 벡터를 스무딩하는 방법<sup>7)</sup>으로 처리한 스무딩된 움직임 벡터들을 이용한 경우이다. BMA의 움직임 추정 방법은 전역 탐색법을 사용하여 화소 단위가 지로 한다. 블록의 크기는 8×8과 16×16을 사용한다.



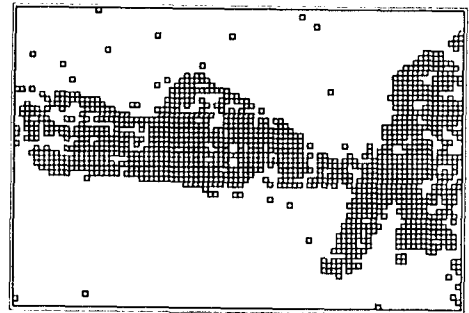
(a)



(b)

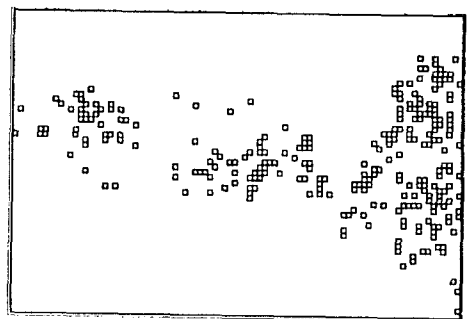
그림 5. 실험에 사용한 원영상 (a) 'Football' 영상, (b) 'Flower Garden' 영상

Fig. 5. Original Images: (a) 'Football' image, (b) 'Flower Garden' image.



(a) 탐색 영역이 이동된 블록

(a) Blocks with moved search region.

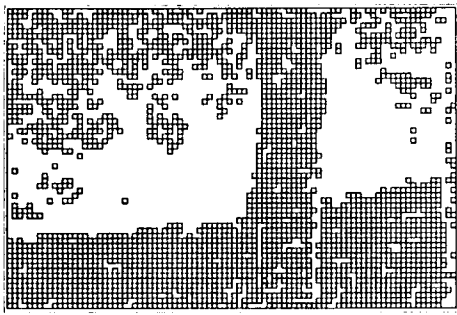


(b) 예측 오차가 개선된 블록

(b) Blocks with improved prediction error

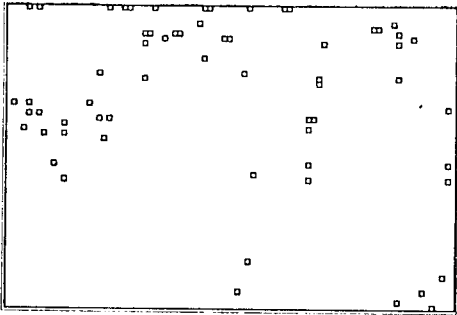
그림 6. 'Football' 영상의 처리 결과

Fig. 6. Results of 'Football' image.



(a) 탐색 영역이 이동된 블록

(a) Blocks with moved search region.



(b) 예측 오차가 개선된 블록

(b) Blocks with improved prediction error

그림 7. 'Flower Garden' 영상의 처리 결과  
Fig. 7. Results of 'Flower Garden' image.

그림 5는 실험에 사용한 원영상이다. 그림 6은 'Football' 영상에서 탐색 영역이 이동된 블록과 예측 성능이 개선된 블록을 표시한다. 그림 6의 (a)는 탐색 영역이 이동된 블록이고, (b)는 탐색 영역이 이동된 블록에서 예측 성능이 개선된 블록만을 표시한다. 그림 7은 'Flower Garden' 영상을 처리한 결과이다. 그림 6와 그림 7에서 탐색 영역이 이동된 블록을 보면 움직임이 존재하는 부분에서 탐색 영역의 이동이 일어났음을 알 수 있다.

표 1. 재구성된 영상의 PSNR(dB)  
Table 1. PSNR of reconstructed images(dB).

영상	방법	전역 탐색	Telescopic 탐색		적응적 탐색 영역 이동	
			원래의 MV	스무딩된 MV	원래의 MV	스무딩된 MV
Football	8×8	27.694	27.858	27.898	28.161	28.149
	16×16	25.028	25.149	25.192	25.476	25.458
Flower Garden	8×8	29.879	29.893	29.891	29.952	29.942
	16×16	28.046	28.051	28.051	28.077	28.075

표 1에 15프레임을 처리한 결과의 평균 PSNR을 비교하였다. 결과적으로 탐색 영역을 이동시킨 경우가 화질이 향상되며, telescopic 탐색 개념을 이용하여 탐색 영역을 이동시킨 경우보다 제안한 방법이 우수한 성능을 나타낸다. 스무딩된 움직임 벡터를 탐색 영역의 이동에 사용하는 경우에 있어서는, 스무딩함으로써 부정확한 움직임 벡터들을 교정해 주기 때문에 더 많은 화질 향상이 이루어질 것으로 기대되었지만 결과는 반드시 그렇지 않았다. 이는 스무딩된 움직임 벡터도 정확히 실제 움직임을 표현하지 못하는 경우에서 발생하는 것이라 생각한다. 또한 16×16의 블록 크기보다는 8×8일 경우가 화질 개선 효과가 더 크다. 이는 작은 블록일수록 블록내의 모든 화소가 같은 움직임을 갖는 경우가 빈번히 존재하므로 해당 움직임 벡터가 신뢰성이 크기 때문이다. 즉 탐색 영역을 이동시킬때 비교적 정확한 이동량을 산출하는 근거가 된다.

표 2. 각 프레임에서 예측 성능의 개선이 있는 블록

Table 2. Block number with improved performance in each frame.

	'Football' 영상		'Flower Garden' 영상	
	8×8	16×16	8×8	16×16
프레임 4	404	115	110	14
프레임 5	367	985	90	16
프레임 6	371	97	102	11
프레임 7	344	81	99	10
프레임 8	398	105	97	5
프레임 9	428	121	94	10
프레임 10	417	120	89	12
프레임 11	495	139	80	8
프레임 12	555	161	81	9
프레임 13	661	195	88	14
프레임 14	716	202	99	12
프레임 15	657	202	88	11

표 2는 각 프레임에서의 예측 성능이 개선된 블록 수를 표시한다. 'Football' 영상의 처리 결과, 8×8 블록일 경우에는 전체 5400개의 블록중에서 프레임당 평균적으로 484개의 블록에서 예측 성능의 개선이 발생하였으며, 16×16 블록일 경우는 전체 1350

개의 블록중에서 프레임당 128개 블록에서 예측 성능의 개선이 발생하였다. 'Flower Garden' 영상은 8 8 블록일 경우에 프레임당 평균적으로 93개 블록에서 예측 성능의 개선이 발생하였고, 16×16 블록은 프레임당 약 11개의 블록에서 예측 성능의 개선이 발생하였다. 'Football' 영상이 'Flower Garden' 영상보다 움직임이 크기 때문에 탐색 영역을 이동시키는 효과가 많이 나타난다.

BMA에 의한 움직임 벡터는 물체의 움직임을 정확히 표현해 주지 못하므로 움직임 벡터에 의한 탐색 영역의 이동이 때로는 부정확한 결과를 초래할 수 있다. 이런 경우에는 예측 성능의 감소가 발생한다. 실험 결과 예측 성능이 감소하는 블록이 'Football' 영상에서는 8 8 블록일 경우에 프레임당 310개, 16×16 블록일 경우에 67개 정도가 발생하고 있다. 'Flower Garden' 영상에서는 8×8 블록일 경우에 22개, 16×16 블록일 경우에 7개 정도가 발생한다.

제안한 방식은 프레임 전체적으로 보면 화질 개선 효과가 미소하게 나타나는데, 이는 탐색 영역 이동에 의해 예측 성능이 개선되는 블록의 수가 전체 블록수에 비해 매우 작기 때문이다. 그러나 성능이 개선된 블록만을 대상으로 예측 성능의 정도를 산출하면 그 효율은 크다고 할 수 있다. 또한 움직임이 전체적으로 많은 영상을 대상으로 적응적으로 탐색 영역을 이동시키는 방법을 적용하면 효율이 확대될 수 있다.

## V. 결론

본 논문에서는 탐색 영역을 효율적으로 이동시켜 탐색 영역을 확대시키는 효과를 얻으며 나아가서 정확한 움직임 추정에 의하여 예측 오차를 감소시키는 것을 목표로 하고 있다. 이와 같은 접근은 기존의 움직임 추정보다 다음과 같은 잇점이 있다. ① 움직임 벡터의 크기를 줄일 수 있으므로 전송 비트량을 줄일 수 있다. ② 탐색 시간과 계산량을 줄일 수 있다. ③ 같은 크기의 탐색 영역을 갖는다면 예측 오차를 줄일 수 있어 전송율을 감소시킬 수 있다.

본 논문에서는 탐색 영역의 크기를 일반적인 BMA와 같게 하고 예측 오차를 줄이는데 중점을 두었다. 모의 실험 결과 같은 크기의 탐색 영역을 갖는 경우

와 비교해서 움직임이 큰 블록에 대해서는 예측 오차를 줄일 수 있었다. 앞으로 연구로, 더욱 정확한 움직임 벡터를 구하는 방법이 필요하리라 본다. 정확한 예측이 가능하다면 정확한 탐색 영역의 이동이 이루어질 수 있고, 이동된 탐색 영역을 근거로 했을 때 예측 오차를 더 감소시킬 수 있다.

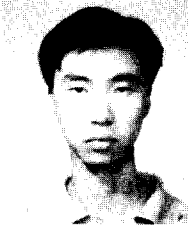
## 參考文獻

- [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] ISO/IEC JTC1/SC29/WG11 MPEG94/N0702, *Generic Coding of Moving Pivtures and Assiciated Audio*, Mar 1994.
- [4] *Grand Alliance HDTV System Specification*, Advisory Committee on Advanced Television Services Experts Group, Feb. 1994.
- [5] ISO-IEC/JTC1/SC29/WG11 MPEG 92/175, "Full search, telescopic search and half pel refinement," Mar. 1992.
- [6] A. Puri and R. Aravind, "On comparing motion - interpolation structures for video coding," in *Proc. Visual Communications and Image Processing*, SPIE vol. 1360, 1990, pp. 1560-1571.
- [7] 김진태, 김기현, 최중수, "움직임 벡터의 신뢰성 향상을 위한 스무딩 방법에 관한 연구", 전자공학회 논문지, 제31권 B편 제5호, pp. 108-116, 1994.
- [8] 이석호, 장태규, 최중수, "적응적인 탐색영역을 이용한 BMA 알고리즘", 제4회 신호처리 합동학술대회 논문집, pp. 398-401, 1991.

---

 著 者 紹 介
 

---



李錫鎬(正會員)

1967年 3月 13日生. 1989年 2月  
 중앙대학교 전자공학과 졸업(공학사). 1992年 2月 중앙대학교 대학원 전자공학과 졸업(공학석사). 1992年 3月 ~ 현재 한국통신. 주 관심 분야는 멀티미디어 등임.

金珍台(正會員) 第 31卷 B編 第 1號 參照  
 현재 중앙대학교 시간강사

•

崔宗秀(正會員) 第 28卷 B編 第 5號 參照  
 현재 중앙대학교 전자공학과 교수