

# 적응 탐색 영역을 가지는 고속 움직임 추정 알고리즘

정 성 규<sup>†</sup>, 정 차 권<sup>‡</sup>, 조 경 록<sup>†</sup>

<sup>†</sup> 충북대학교 정보통신공학과 통신소재연구소

<sup>‡</sup> 호서대학교 제어계측공학과

## Fast Motion Estimation Algorithm with Adjustable Searching Area

*Sung Gyu Jung<sup>†</sup>, Cha Keon Cheong<sup>‡</sup>, and Kyoung Rok Cho<sup>†</sup>*

<sup>†</sup> Department of Computer & Communication Engineering School, Chungbuk Nat'l Univ.

<sup>‡</sup> Dep. of Control & Instrumentation eng. Hoseo Univ.

### Abstract

A new motion estimation algorithm with lower computational complexity and good image quality when compared to the FBMA will be presented in this paper. In the proposed method, by considering the relation between neighboring blocks, the searching area in the algorithm is adjustable according to mean absolute difference of the block. By the computer simulation the computation amount of the proposed method can be greatly decreased than that of the FBMA and the good result of the PSNR can be attained

### 1. 서 론

통신 매체의 발달로 고속으로 정보를 전달하는 것이 가능하게 되었으며, 동영상 정보를 전송하기 위한 표준안으로 MPEG(Motion Picture Expert Group)이 표준화되어 실용화되고 있다. 동영상은 대체로 움직임 보상

(motion compensation)을 이용한 하이브리드 코딩으로 높은 데이터 압축을 달성하고 있다. 이 방법에 의한 데이터의 압축은 이전 프레임을 이용하여 움직임 추정 및 보상을 수행하고, 이때 추정된 움직임 벡터(motion vector)에 의해서 보상된 영상과 원 영상과의 차 신호를 부호화한다.<sup>[1]</sup> 움직임 추정은 크게 화소 순환 알고리즘(pixel recursive algorithm: PRA)과 블록 정합 알고리즘(block matching algorithm: BMA)으로 나누어진다.<sup>[2][3]</sup> 블록 정합 알고리즘은 화소 순환 알고리즘에 비하여 수행시간이 적게 소요되고 하드웨어 구현이 용이하기 때문에 움직임 보상 부호화에 널리 사용되고 있다.

블록 정합 알고리즘에서는 한 블록대의 모든 화소들이 같은 방향으로 평행 이동한다는 것을 가정하여 움직임 추정을 하는 방법으로서, 일정한 탐색 범위에서 모든 화소에 대한 블록을 순차적으로 비교해 보는 전 탐색 블록 정합 알고리즘(full search block matching algorithm: FBMA)을 기준으로 하고 있으며, 이는 최적의 움직임 벡터를 추정할 수 있지만 계산량이 많은 단점을 갖는다. 이와 같은 단점을 줄이기 위해 탐색 범의 수를 줄임으로써 움직임 벡터를 보다 고속으로

로 추정할 수 있는 방법들이 연구되었다.<sup>[1][2]</sup>

대표적인 고속 BMA 방법으로는 2차원 로그 탐색 (two dimensional logarithm search: 2-D LOG), 3단계 탐색 (three step search: TSS) 알고리즘, 교차 탐색 (cross search) 알고리즘, 새로운 3단계 탐색 알고리즘 (new three step search: NTSS) 등이 있다.<sup>[1][2]</sup>

3단계 탐색 알고리즘<sup>[6]</sup>은 움직임 벡터를 추정하는 과정에서 극부최소에 빠지는 단점이 있으며, 새로운 3단계 탐색 알고리즘은 움직임이 작은 영상인 경우에는 비교적 정확하게 움직임 벡터를 추정하지만 움직임이 빠른 영상인 경우에는 전역 최소(global minimum)에 빠져서 부정확한 움직임 벡터를 추정하는 단점이 있다.

이 방법들은 고속으로 움직임 벡터를 추정하기 위해서 전 탐색 블록 정합 알고리즘보다 탐색점의 수를 줄여 근사 움직임을 찾은 후 미세 움직임을 단계적으로 찾은 것으로 전 탐색 블록 정합 알고리즘 보다 계산량은 많이 줄일 수 있으나, 움직임 벡터의 추정이 부정확하게 되므로 물체의 보다 정확한 움직임을 표현하기 위한 방법의 도입이 필요하다.

본 논문에서는 움직임 벡터를 정확하게 추정하면서 계산량을 감소시킬 수 있는 고속 블록 정합 알고리즘을 제안한다. 제안한 방법에서는 정합의 척도로 평균 절대 오차 (mean absolute difference: MAD)를 사용하고, 각 블록의 움직임 정도에 따라 탐색 영역을 변화시켜 움직임 벡터 추정을 행한다. 또한, 제안 방법의 성능을 평가하기 위해서 두 가지 동영상에 대하여 컴퓨터 시뮬레이션을 수행해서 제안 방법의 성능을 평가한다. 이 결과로부터  $16 \times 16$  블록에서 제안 방법은 움직임 추정 방법 중 최적인 전 탐색 블록 정합 알고리즘에 비해 계산량을 81.89% 내지 82.38% 줄일 수 있으며, 비교적 양호한 PSNR을 얻을 수 있음을 보인다.

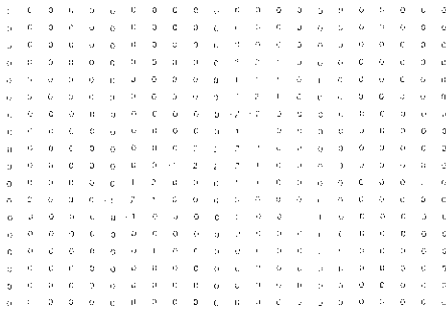
## 2. 제안한 적응 탐색 영역 알고리즘

블록 정합 알고리즘은 현재 프레임  $f_i$ 의 블록  $(i,j)$ 를 이전 프레임  $f_{i-1}$ 에 있는 블록  $(i,j)$ 와 정합을 할 때 평균 절대 오차는 항상 탐색 영역의 모든 탐색점에 대하여 평가를 해야만 움직임 추정이 정확하게 된다. 이와 같이 모든 탐색점에 대하여 움직임 벡터를 추정하면 그 계산량은 방대하게 된다. 또한 고속으로 움직임 벡터를 추정하기 위해서 탐색점 수를 너무 적게 하여 블록 정합 알고리즘을 수행하게 되면 극부 최소에 빠져서 부정확한 움직임 벡터를 추정하게 되어 복원 화상에서 화질이 저하된다.

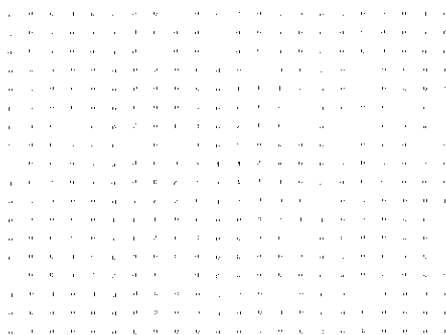
본 논문의 제안 방법은 기존의 전 탐색 블록 정합 알고리즘에서 요구되는 방대한 계산량을 줄임과 동시에 PSNR의 저하를 적게하도록 하기 위해 영상에서 블록의 움직임 정도를 판단하고 이를 기준으로 적당한 탐색 영역을 할당해서 움직임 벡터를 추정한다.

일반적인 화상에서 움직이는 물체는 프레임 내에서 몇몇 블록들에 의해 생성된다. 그러므로 이웃하는 블록의 움직임 벡터 사이에는 매우 유사한 관계가 있다. 즉, 이웃하는 블록의 움직임 벡터 상관성은 매우 높고 움직임 추정 과정에서의 효율성을 향상시키는 데에 있어 중요한 정보이다. 그림 1은 영상 "Salesman"의 연속하는 두 프레임의 움직임 추정 결과이다. 그림 1(a)와 그림 1(b)에서 각각 동일한 위치의 숫자가 움직임 벡터를 구성한다. 프레임 크기가  $352 \times 288$ 이고 블록 크기가  $16 \times 16$ 이므로 움직임 벡터의 수는  $(22 \times 18)$ 이 된다. 앞 관촬로 인해, 이웃하는 블록의 움직임 벡터에 따라 현재 블록의 움직임 벡터를 바로 예측할 수 있다는 것을 쉽게 알 수 있다. 또한 움직임 벡터의 예측으로 탐색 영역을 움직임이 이동하는 방향에 대해서만 고려하여도 움직임 추정 과정에서는 크게 문제가

없다. 즉, 움직임 벡터의 방향에 따라 탐색 영역을 적당히 변화시켜 계산량 및 비교적 정확한 움직임 벡터를 얻을 수 있는 것이다.



(a)



(b)

그림 1. 이웃하는 블록 사이의 움직임 벡터 관계

- (a) 움직임 벡터의 x 성분
- (b) 움직임 벡터의 y 성분

FBMA 방식은 블록을 중심으로 x와y 방향으로 모두 같은 크기의 탐색 영역을 가진다. 하지만 제안한 방법은 블록 중심으로부터 각 방향이 모두 적당한 크기의 범위를 갖는다. 또한 움직임 벡터에 따라 한 방향에 대해서만 고려한다. 제안한 적응 탐색 영역 알고리즘(ABMA)에서 현재 블록의 탐색 영역은 단지 전 블록의 움직임 벡터에 의존한다. 만약 전 블록의 움직임 벡터가 전 블록의 탐색 영역보다 작으면 현재 블록의 움직임 벡터는 또한 작아 될 것이다. 반대로 전 블록의 움직임 벡터가 전 블록의 최대 탐색 영역에 가깝다면 현재 블록의 움직임 벡터 또한 최대 탐색 영역에 가까울 것이므로 현재 블록의 탐색 영역도 최대 탐색 영역을 가져야 할 것이다.

탐색 영역의 설정은 이웃 블록의 움직임 벡터 x,y값에 따라 이루어진다. 즉 x 값에 따라 각각의 th값을 두어 양의 방향(x값이 +)일 때는 음의 방향의 탐색 영역을 줄이고 음의 방향(x값이 -)일 때는 양의 방향의 탐색 영역을 줄여 탐색 영역을 적응시킨다. 여기서 th는 문턱값을 나타내며 y도 탐색 영역을 적응시키는 방법은 x와 같다.

움직임 벡터의 x(y)값이 0이면 현재 블록의 최대 탐색 영역은 +2가 되며 이 값까지는 하나의 탐색 영역을 증가시켜 움직임 벡터를 중심으로 한 적응 탐색 영역을 가지게 하므로써 계산량의 낭비를 줄여보도록 한다. 표 1은 각 움직임 벡터 x(y)값에 따른 탐색 영역을 보인다.

MV(x y):	0	1	2	3	4 ~ 6
		1	2	3	4 ~ 6
탐색 영역	+2	0 ~ 3	0 ~ 4	0 ~ 5	0 ~ 6
		3 ~ 0	4 ~ 0	5 ~ 0	6 ~ 0

표 1. 움직임 벡터에 따른 탐색 영역 변화  
MV: 이웃 블록의 움직임 벡터의 크기  
최대 탐색 영역의 크기: 6

제안한 방법을 적용하여 예측된 탐색 영역은 움직임이 작은 경우에 탐색 영역의 크기가 많이 줄어들어 불필요한 계산에 따른 낭비를 막을 수 있으며 움직임이 큰 경우엔 탐색 영역의 크기를 최대 위치까지 확대하여 정확한 움직임 추정을 할 수 있다.

### 3. 모의 실험 결과

본 실험은 352 × 288 크기의 "Foreman"과 "Carphone" 동영상에 제안한 알고리즘을 적용하였다. 블록의 크기는 16 × 16로 정하였고, FBMA의 경우 최대 탐색 영역의 크기를 6으로 사용하여 움직임 추정을 수행하였다.

제안 방법에서는, 탐색 영역을 예측한 다음 예측되어진 탐색 영역의 모든 점에서 정합을 수행하였다. 그림 2에는 "Foreman" 동영상 30 프레임에 대해 실험하였을 때 FBMA와의 블록 정합수의 비교를 도시하였다. 또한 그림 3과 그림 4에는 "Foreman" 동영상 30프레임과 "Carphone" 동영상 30프레임에 대해 실험하였을 때 구한 PSNR값을 도시하였다.

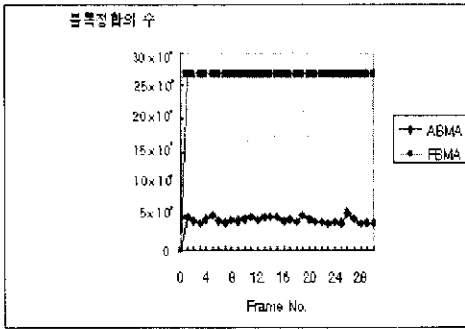


그림 2. 전역 탐색 및 적응 탐색의 블록 정합 수  
 “Foreman” 연속 영상에 대한 평균 블록 정합 수 :  $4.72 \times 10^4$

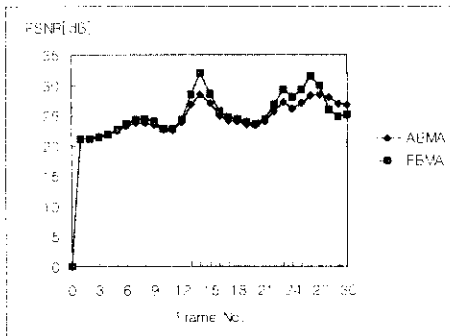


그림 3. “Foreman” 연속 영상에 대한 PSNR  
 평균 PSNR (ABMA:24.76, FBMA:25.40)

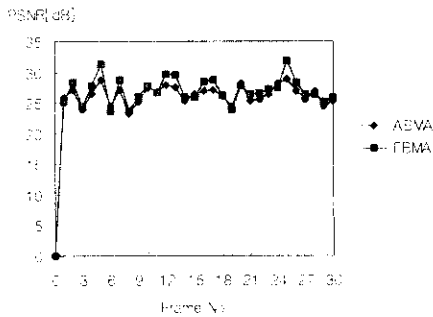


그림 4. “Carphone” 연속 영상에 대한 PSNR  
 평균 PSNR (ABMA:26.31, FBMA:27.02)

두 영상의 경우 모두 제한된 방법을 적용했을 때 PSNR값은 고정된 탐색 영역의 FBMA를 사용한 것보다 높거나 거의 같은 값을 나타내고 있음을 알 수 있다.

그림 3과 그림 4은 “Foreman” 동영상과 “Carphone” 동영상에 대해 FBMA 방법과 제한된 방법을 사용하였을 때의 각 PSNR값을 비교

하였다. 제한된 방법을 사용했을 경우 FBMA 방법과 비슷한 값을 구할 수 있었으며, 계산량에 있어서는 80%정도 감소했음을 알 수 있다. FBMA는 움직임이 적은 부분에 대해서도 고정된 탐색 영역을 사용하므로 불필요한 영역에 대해 움직임 추정을 수행하게 되지만 제한된 방법은 탐색 영역의 적응적 축소를 통해 계산량을 크게 감소시킬 수 있다.

#### 4. 결 론

동영상은 움직임 추정에 대해 많은 연구 논문이 발표되고 있고 움직임을 추정할 때 계산량을 줄이고 있지만 FBMA에 비해 정확하지 못한 움직임 추정으로 인해 영상의 화질이 상당히 저하되는 단점이 있다. 본 논문에서는 적응 탐색 영역을 사용하여 영상의 화질을 유지하였을 뿐만 아니라 상당한 계산량의 감소도 얻을 수 있었다.

#### 참 고 문 헌

- [1] H.G. Musmann, P.Pirch and H.J.Grallert, "Advanced in picture coding," *Proc. of IEEE*, vol. 73, no. 4, pp.523-548, Apr.1985
- [2] A. K. Jain, "Image data compression: A review,"*Proc. of IEEE*, vol. 69, no. 3, pp.349-389, Mar. 1981.
- [3] T. Koga et al., "Motion-compensated interframe coding for video conferencing," *in Proc. Nat. Telecom. Conf.*,pp.G 5.34-G 5.35, Nov. Dec. 1981.
- [4] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. Circuit and Systems for Video Technology*, vol. 6, pp. 148-157, Apr 1993.
- [5] M. Ghanbari, "The cross search algorithm for motion estimation," *IEEE Trans. on Commun.*, vol. COM-38, no. 7, pp. 950-953, July 1990.
- [6] R. Li, B. Zeng, and M. L. Liou,"A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuit and Systems for Video Technology*, vol. 4, no. 4, pp. 138-142, Aug. 1994.