

Block의 평균과 분산을 이용한 Block Matching Algorithm

김진업, 고종석, 김계균  
한국과학기술원 전기및전자공학과

A Block Matching Algorithm Using the Block Mean and Variance

Jin-up Kim, Jong-seog Koh, Jae-kyoon kim  
Dept. of Electrical Engineering, K A I S T

Abstract

An improved method for the interframe motion estimation is proposed. This is a sort of menu vector search method using the mean and variance of the blocks in previous-frame search area and those of the current coding block.

Computer simulation shows that the method is more effective and computationally simpler than other methods, i.e., Three Step Search and 2-D Logarithmic Search.

1. 서론

영상 데이터의 효과적인 압축을 위해서는 temporal domain에서의 redundancy를 줄이는 방식이 필요하다. 이러한 방식에는 영상에서의 움직임 보상을 주는 MCC (Motion Compensated Coding)가 데이터 압축 성능이 우수하다. 이러한 MCC는 물체의 움직임을 검출하는 방법에 따라 PRA(Pel Recursive Algorithm)[1]와 BMA(Block Matching Algorithm)[4,5,6,7]로 나눌 수 있다.

본 연구에서는 BMA 방법중 block의 평균과 분산을 이용하여 이동 벡터를 검출하는 방법에 관하여 연구하였다.

본 논문의 구성은 2절에서 BMA 에서 고려되어야 할 사항을 논하고, 3절에서 이동 벡터를 검출하는 제안된 방법을 논하며, 4절에서 시뮬레이션 방법 및 결과물, 5절에서 전체적인 결론 및 검토를 하였다.

2. BMA(Block Matching Algorithm)에서 고려할 사항

BMA 라 함은 입력 영상을 작은 block으로 나눈 다음, 그림 1 과 같이 과거 frame 에 적당한 크기의 search area 를 두고 그 search area 내에서 현재 coding 하고자 하는 block과 가장 match가 잘되는 block을 찾아서 이동 벡터를 검출하는 방법을 말한다. 여기에서 고려되어야 할 사항은 search area의 크기 및 block의 크기, matching criterion, 그리고 search 방법 등이다.

한 frame 사이(1/30 초)에 영상내의 물체가 변화 할수있는 최대 변위를 p 라하고 block의 크기가 M\*N 일경우 search area의 크기는 (M+2p)\*(N+2p)로 정해진다. 이 search area의 크기가 클 수록 움직임이 큰 경우도 보상에 줄 수 있는 반면, 그 변위 즉, 이동 벡터를 검출 하는데 있어서 계산량이 증가하게 된다. 따라서 이동 벡터를 검출할때의 계산량과

운동 보상의 영역을 고려한 적절한 search area가 결정 되어야 한다.

또한 block의 크기가 클 수록 block간의 mismatch의 정도가 커지므로 data 압축 효과는 작아지나 block당 전송해야 하는 이동 벡터의 양이 작아지므로 이러한 side-information 과 data 압축 효과를 고려한 적절한 block의 크기가 결정되어야 한다. 일반적으로 block의 크기는 8\*8의 크기가 적절함이 알려져 있다[2].

그리고 matching criterion은 현재 block과 과거 frame에서 match 하고자 하는 block과의 matching의 정도를 나타내는 척도로서 NCCF(Normalized Cross Correlation Function), MSE(Mean Square Error), MAE(Mean of the Absolute Error)등이 있다[3].

BMA에서 block의 크기를 M\*N으로하고 최대 변위를 p 이라고 가정하면 이동 벡터가 가질 수 있는 경우의 수는 (2\*p+1)\*(2\*p+1)=169 가 되어 상당히 많은 계산을 요구한다. 따라서 이러한 계산량을 줄이고 실시간 처리를 용이하게 하는 방법들이 연구되어 왔다. 이러한 방법들로는 2-D Logarithmic Search[4], Three Step Search[5], Modified Direction Search[6], Menu Vector Search[7] 등이 있다.

3. 평균과 분산을 이용한 이동 벡터 검출 방법

기존의 BMA 에 의한 이동 벡터 검출 방법은 하드웨어 관점에서 sequential step 수가 문제가 된다. 즉, sequential step 수가 많을 경우 parallel processing이 불가능 하기 때문에 time delay 가 많아지게 되므로 실시간 처리에 적합하지 않다. 따라서 참고문헌[7]의 menu vector search 방법이 1 step 뿐이므로 실시간 및 하드웨어 관점에서 유리하다. 그러나 항상 고정된 위치에서만 search가 수행되므로 그 성능이 떨어진다. 본 연구에서는 이러한 menu vector search 방법에서 menu vector를 선정하는데 있어서 평균과 분산을 이용하므로써 그 성능을 개선 하였다.

(1) Block Distortion 과 Correlation 과의 관계

a(i,j)를 현재 coding 하고자하는 block의 (i,j)위치의 pel, b(i,j)를 이전 frame 내의 (i,j) 위치의 pel, 그리고 block의 크기를 M\*N 이라고 할 경우, 이전 frame 내의 block이 현재의 block 보다 (p,q) 만큼 변위 되었을 때 두 block 사이의 distortion E<sub>pq</sub> 는 MSE로 정의하며 다음과 같다.

$$E_{pq} \triangleq \frac{1}{M*N} \sum_{i=1}^M \sum_{j=1}^N [a(i,j) - b(i-p,j-q)]^2 \dots \dots (1)$$

이때 search area 의 모든 block에 대해 위의  $E_{pq}$ 가 최소가 되는 block 의 벡터 (p,q)가 이동 벡터가 된다. 여기서 match 되는 두 block 의 correlation  $\rho_{pq}$  는 다음과 같다.

$$\rho_{pq} \triangleq \frac{\frac{1}{M*N} \sum_{i=1}^M \sum_{j=1}^N [a(i,j) \cdot b(i-p, j-q)] - m_a \cdot m_b}{\sigma_a \cdot \sigma_b} \dots (2)$$

$$m_a = \frac{1}{M*N} \sum_{i=1}^M \sum_{j=1}^N [a(i, j)]$$

$$m_b = \frac{1}{M*N} \sum_{i=1}^M \sum_{j=1}^N [b(i-p, j-q)]$$

$$\sigma_a^2 = \frac{1}{M*N} \sum_{i=1}^M \sum_{j=1}^N [a(i, j) - m_a]^2$$

$$\sigma_b^2 = \frac{1}{M*N} \sum_{i=1}^M \sum_{j=1}^N [b(i-p, j-q) - m_b]^2$$

식 (1) 과 식 (2) 로부터 다음식을 구할 수 있다.

$$E_{pq} = \sigma_a^2 + \sigma_b^2 + (m_a - m_b)^2 - 2 \rho_{pq} \sigma_a \sigma_b \dots (3)$$

$$E_{pq} = (\Delta\sigma)^2 + (\Delta m)^2, \text{ if } \rho_{pq} = 1$$

여기서  $\Delta\sigma \triangleq \sigma_a - \sigma_b$

$\Delta m \triangleq m_a - m_b$  이다.

이 식에서 물체가 sample 간격의 정수배 만큼 평행이동했고, 그 벡터가 (p,q) 라고 한다면 두 block 사이의 correlation 은 1 이 되어야 하고 distortion 은 0 이 되어야 한다. 그러나 실제의 영상에서는 그 움직임이 평행이동 만이 존재하지 않고 회전운동, 병진운동 등 3차원의 복잡한 움직임을 가지며 또한 spatial noise, temporal noise 등의 영향으로 그 distortion 은 0 이 되지 않는다. 따라서 평균과 분산 만으로 정확한 이동 벡터를 찾는 것은 불가능하나 적당한 criterion 을 선정하면 정확한 이동 벡터와 유사한 menu vector 를 선정 할 수 있다.

### (2) 이동 벡터의 선정 방법

Search area 내의 match 하고자 하는 하나의 변위 벡터 (p,q) 가 주어지면 식 (3) 의 모든 값은 결정지어진다. 여기서 평균과 분산은 coding block 이 결정되기 이전에 계산이 수행될 수 있는 값이나,  $E_{pq}$  와  $\rho_{pq}$  는 coding block 이 결정 되어야만 계산이 수행되어 진다. 따라서 모든 (p,q) 조합에 의한  $E_{pq}$  와  $\rho_{pq}$  를 구하기는 상당한 계산량이 요구된다. 그러므로 이러한 막대한 계산을 피하기 위하여 다음과 같은 방법으로 menu vector 를 선정하여 이동 벡터를 결정한다.

현재의 block 과 이전 frame 의 search area 내에서 match 하고자 하는 하나의 변위 벡터 (p,q) 가 주어졌을 때 변위된 block 과의 distortion 은 식 (3) 에 의해 구해질 수 있다. 이 식에서 각 block 의 평균과 표준편차는 서로의 block 에 관계없이 계산될 수 있으므로 이들을 상수로 취하며, 서로의 block 에 관계되어 계산되는 값,  $E_{pq}$  와  $\rho_{pq}$  를 변수로 취하면 두 block 의 distortion 과 correlation 은 그림 2 와 같은 관계를 갖는다. 그러나 실제로 match 되는 두 block 이 결정되면  $E_{pq}$  와  $\rho_{pq}$  는 상수값이고 이들을  $e_1, \rho_1$  라고 하면  $e_1$  과  $\rho_1$  이 이루는 점은 직선 AB 상의 한 점C가 된다. 이러한  $e$  과  $\rho$  의 관계를 나타내는 직선 AB 는 match

하고자 하는 때 block 마다 생기며 그 직선상의 어느 한 점에서 두 block 사이의 실제의 distortion 과 correlation 이 결정된다.

일반적으로 임의의 두 block 이 서로 matching 이 잘 될수록 두 block 사이의 correlation 은 1에 가까워지고 distortion 은 0 에 가까워진다. 따라서 그림 2 에서 결정되는 점C는 점A에 근접해지며 점A 역시 0 에 가까워진다. 그러므로 best match 의 경우에 두 block 의 correlation 을 1로 근사화시킬 수 있으며 식 (3)으로 부터 distortion 을 최소화 시키는 벡터, 즉,  $(\Delta\sigma)^2 + (\Delta m)^2$  이 최소가 되는 벡터를 결정할 수 있다. 그러나 실제적으로 두 block 의 correlation 은 1 이 아니므로  $(\Delta\sigma)^2 + (\Delta m)^2$  이 최소가 되는 벡터가 실제의 이동 벡터일 수는 없으나, 그 값이 작을 수록 실제의 이동 벡터일 확률이 통계적으로 크다. 이러한 통계적 특성을 이용하여 모든 (p,q) 조합에 대해  $(\Delta\sigma)^2 + (\Delta m)^2$  을 구한후 그 값이 작은순으로 일정 개수 만큼의 (p,q) 조합을 menu vector 로 취한다. 여기서 모든 (p,q) 조합을 menu vector 로 취하면 exhaustive search 의 경우와 같다.

본 연구에서는 하드웨어 등을 고려하여 menu vector 의 수를 8, 10, 16, 25 개로 제한하였고 기존의 방법과 비교하였다.

### 3. 시뮬레이션 방법 및 결과

BMA 는 두 영상으로 부터 물체의 이동 정보를 찾아내어 그 정보로 부터 현재의 block 을 prediction 하므로써 prediction error 를 줄이는 데 있다. 따라서 그의 성능을 측정하기 위해서는 prediction error 의 entropy 와 SNR (Signal to Noise Ratio) 이 사용 된다. 여기서 entropy란 그 데이터를 표시할수있는 최소 bit 수를 말하고, SNR은 입력 영상의 최대치의 제곱과 error power의 비로서 다음과 같이 정의 된다.

$$\text{entropy} \triangleq \sum_{i=-255}^{255} P_i \cdot \log_2 \frac{1}{P_i} \dots (4)$$

$$\text{SNR} \triangleq 10 \cdot \log_{10} \frac{255^2}{E[(X-\bar{X})^2]} \dots (5)$$

여기서  $P_i = \text{Prob.}[\text{Prediction error} = i]$

$X$ : 원 영상 신호

$\bar{X}$ : 예측 영상 신호 이다.

또한 BMA의 성능만을 측정하기 위하여 그림 3 과 같은 quantizer를 포함하지 않은 system을 구성하였고 data base는 "Cronkite" 영상을 사용하였다. "Cronkite" 영상의 크기는 256\*256 이고 pel 당 8bit로 sample되어 있으며 16 frame의 연속 영상을 갖는다.

성능비교의 대상으로 현재까지의 이동 벡터를 찾는 방법중 좋은 성능을 갖는 2-D Logarithmic Search, Three Step Search 방법, 그리고 BMA에서 성능의 상한선인 exhaustive search 방법을 비교하였다.

그림 4 에 기존의 방법과 본 연구에서 제안한 방법으로 10개의 menu vector 를 취하여 이동 벡터를 찾는 경우의 entropy를 각 frame sequence 에 대해 비교하였고, 그림 5 에는 그때의 SNR을 비교하였다. 이 그림으로 부터, 본 연구에서 제안한 방법중 10개의 menu vector 를 갖는 경우가 현재까지 가장 좋은 성능을 갖는 2-D Logarithmic Search, Three Step Search 방법과 entropy, SNR 측면에서 거의 같은 성능을 갖음을 알수있다. 그러나 본 연구에서 제안한 방법은 과거 frame의 평균과 분산 그리고 현재 block 의 평균과 분산으로 부터 search해야할 menu vector 를 선정하기 때문에 이들의 계산은 motion search 과정과 독립적으로 수행될 수 있다. 즉 motion search가 수행되기 이전에 각 평균과 분산은 미리 계산 되어질 수 있다. 따라서 motion search

time 을 줄일 수 있으며 sequential step도 menu vector 선정 단계와 motion search 단계 등 2 step으로 다른 방법보다 작다. 또한, 본 연구에서 제안하는 방법이 기존의 BMA 방법과 같은 성능을 갖는 경우의 search point 수는 10번으로 Three Step Search 경우의 25번 보다 작다.

그림 6 에 본 연구에서 제안한 방법으로 menu vector의 수를 8, 10, 16, 25번으로 정했을 때의 entropy를 각 frame sequence 에 대해 비교하였고 그림 7 에는 그때의 SNR을 비교하였다. 이 그림으로부터 본 연구에서 제안한 방법이 menu vector의 수를 증가시킴에 따라 그 성능이 증가함을 알 수 있다. 즉, menu vector의 수를 조절하므로써 motion estimation의 성능을 조절할 수 있다. 여기에서 graph의 모양이 exhaustive search의 경우와 비슷한 것이 특이하다.

### 5. 결론 및 검토

본 연구에서는 과거 frame의 평균과 분산으로부터 menu vector를 결정하고 그 menu vector로부터 이동 벡터를 결정하는 방법을 제안 하였다. 이 방법은 기존의 menu vector search[7] 방법 보다 많은 2개의 sequential step 수를 갖지만 기존의 menu vector search 방법이 고정된 위치에서만 search를 수행하므로써 생기는 성능 저하를 막을 수 있을 한데, 다음 block의 menu vector를 선택하는 과정과 현재 block의 motion search가 서로 독립적으로 수행될 수 있으므로 이들을 병행처리(parallel processing)할 경우, 1번의 step 시간에 처리될 수 있다. 또한 menu vector의 수를 증가시킴에 따라 그 성능은 BMA에서의 성능의 상한선인 exhaustive search에 가까워진다. 그리고 본 연구에서 제안한 방법은 기존의 BMA방법에서 search area가 커짐에 따라서 발생하는 계산량의 증가에 대한 문제를 극복할 수 있다. 또한 영상 신호의 vector quantization에서 중요한 code book search등 그 적용 범위가 기존의 BMA방법 보다 넓을 것으로 기대 된다.

#### [참 고 문 헌]

- [1]. A.N.Netravali, J.D.Robbins, "Motion-Compensated Television Coding : Part I," BSTJ, Vol.58, No.3, March 1979
- [2]. 정 주 흥, "이동 벡터 검출을 위한 Block Matching 방법에 관한 연구," 한국 과학 기술원, 석사 학위 논문, 1985
- [3]. H.G.Mussman, P.Pirsh, H.G.Grallert, "Advances in Picture Coding," Proceedings of the IEEE, Vol.73, No.4, April 1985
- [4]. J.R.Jain, A.K.Jain, "Displacement Measurement and Its Application in Interframe Image Coding," IEEE trans. on comms., Vol.29, No.12, Dec 1981
- [5]. T.Koga, et al, "Motion Compensated Interframe Coding for Video Conferencing," National Telecommunication Conference, Nov.29-Dec.3, 1981
- [6]. R.Srinivasan, K.R.Rao, "Predictive Coding on Efficient Motion Estimation," IEEE trans. on comms., Vol.33, No.8, Aug 1985
- [7]. Y.Ninomiya, Y.Ohtsuka, "A Motion Compensated Coding Scheme for Television Pictures," IEEE trans. on comms., Vol.33, No.8, Aug 1985

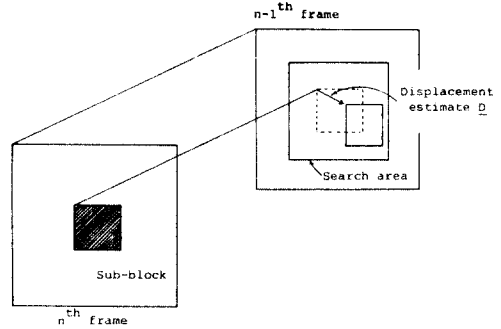


그림 1. BMA에 의한 이동 벡터 검출

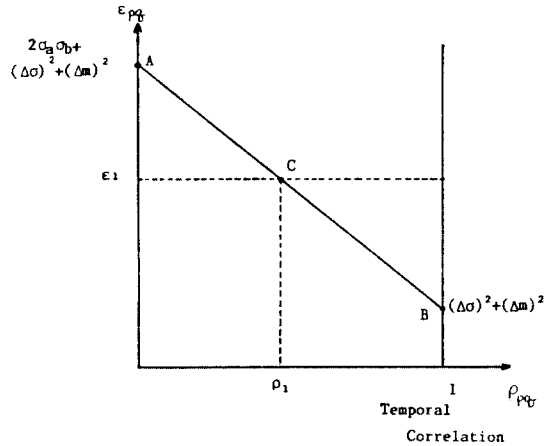


그림 2. 최적의 match를 갖는 두 block사이의 correlation 과 distortion 과의 관계

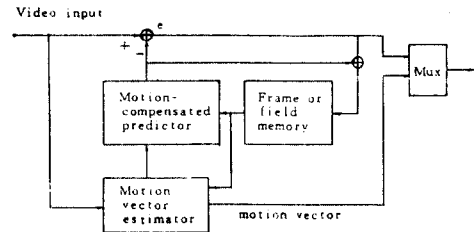


그림 3. Simulation에 사용된 system

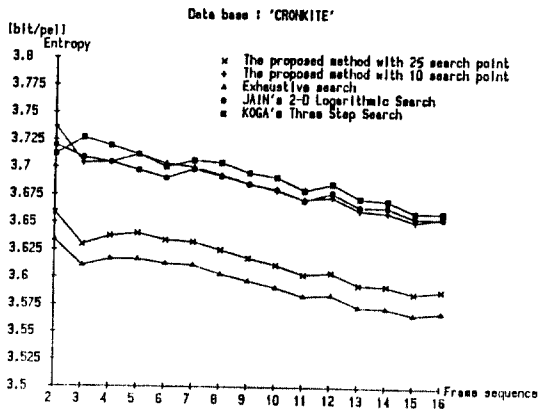


그림 4. 제안된 방법과 기존의 BMA 방법과의 entropy 비교

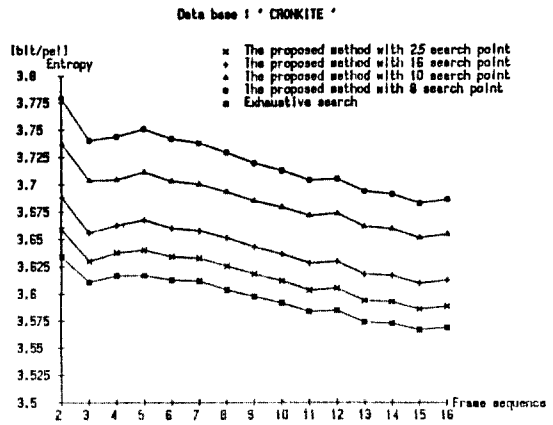


그림 6. Menu vector 수의 변화에 따른 제안된 방법의 entropy 비교

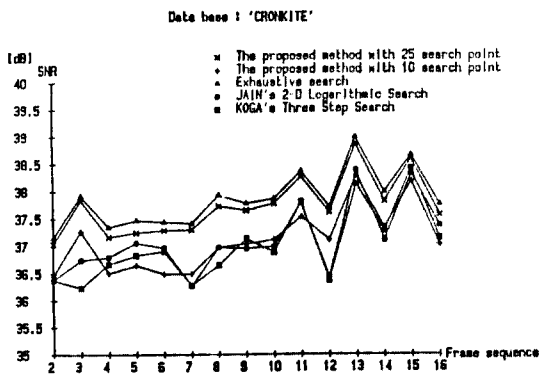


그림 5. 제안된 방법과 기존의 BMA 방법과의 SNR 비교

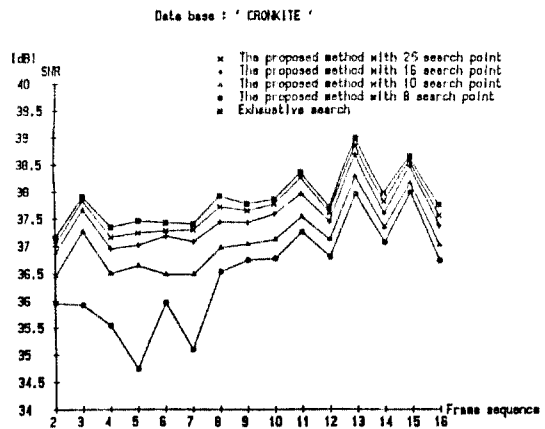


그림 7. Menu vector 수의 변화에 따른 제안된 방법의 SNR 비교