

영상 신호에서 이동 벡터 검출을 위한 BMA의 성능 개선에 관한 연구

정 주홍*, 이 만섭**, 김 재근***

- * 한국전자통신연구소 집속처리연구실
- ** 한국전자통신연구소 광통신시스템연구실
- *** 한국과학기술원 전기및전자공학과

A Study on the Performance Improvement of BMA for Motion Vector Estimation in Image Sequences

J.H. Jung*, M.S. Lee**, J.K. Kim***

- * Communications Processing Section, ETRI
- ** Photonic Systems Section, ETRI
- *** Dept. of Electrical Engineering, KAIST

ABSTRACT

An effective and simple Block Matching Algorithm(BMA) based on "menu vector" search is proposed for motion vector estimation. The proposed BMA is compared with existing "menu vector" search technique in Ref.[6]. In the comparison of its accuracy, the entropy of the prediction error and NMSE(Normalized Mean Square Error) are used as performance measures. The performance of the proposed algorithm is 3dB better than the existing "menu vector" search techniques in NMSE.

1. 서론

영상 신호를 디지털 데이터로 전송하는 경우에 있어서 가장 심각한 문제는 전송 대역폭의 증가이며, 전송 대역폭을 줄이기 위한 여러 가지 영상 데이터 감축 방식이 연구되어 왔다.[1] 디지털 영상 데이터 감축 방식은 구성 형태에 따라 intraframe coding 방식과 interframe coding 방식으로 나눌 수 있는데, 최근에 들어 반도체 기술의 발달로 interframe coding 방식의 하드웨어 구현이 용이해 지면서 이 방면에 많은 연구가 진행되고 있다. interframe coding 중에서는 MCC(Motion-Compensated Coding)가 데이터 감축 성능이 우수한 것으로 알려져 있다. MCC에서 가장 중요한 문제는 이동 벡터를 검출하는 방법이고, 이동 벡터를 검출하는 방법은 PRA(Pel-Recursive Algorithm)와 BMA(Block Matching Algorithm)로 나눌 수 있다.

본 연구에서는 BMA중에서 recursion을 도입한 "menu vector" search를 통하여 보다 효과적으로 이동 벡터를 검출하는 방법에 관하여 연구하였다. 본 논문의 구성은 제1장에서 BMA에 의한 이동 벡터 검출을 논하고, 제2장에서 BMA의 성능 개선 방법을 기술하였고, 제3장에서 시뮬레이션 방법 및 결과를 보였고, 제5장에서 전체적인 검토 및 결론을 맺었다.

2. BMA에 의한 이동 벡터 검출

BMA(Block Matching Algorithm)란 입력 영상을 작은 block으로 나눈 다음 그림.1과 같이 이전 frame에 적당한 크기의 search area를 두고 search area 내에서 가장 matching이 잘된 block을 찾아서 이동 벡터를 검출하는 방법을 말한다.

BMA에서 고려해야 할 사항은 search area의 크기 및 sub-block의 크기, matching criterion, search 방법등이다. 이 중에서 search area의 크기는 한 frame 사이에 변화할 수 있는 최대 이동 벡터를 p라 하면 sub-block의 크기가 $M * N$ 인 경우 search area의 크기는 $(M+2p) * (N+2p)$ 로 정해진다. 그리고 matching criterion은 현재 block과 이전 frame의 search area 내의 block 사이의 matching의 정도를 나타내는 척도로서 NCCF(Normalized Cross Correlation Function), MSE(Mean Square Error), MAD(Mean of the Absolute Frame Difference)가 있다.[2]

BMA에서 sub-block의 크기가 $M * N$ 이고 최대변위 p가 6이라고 가정하면, 이동 벡터의 경우의 수는 $(2p+1)^2 = 169$ 가되어 이러한 계산은 실시간 처리에 불리하므로, 이를 극복 하기 위한 방법들이 제안되었는데, 이러한 방법들로는 2-D Logarithmic Search [3], Three Step Search [4], Modified Conjugate Direction Search [5], "menu vector" Search [6] 등이 있다.

3. BMA의 성능 개선 방법

기존의 BMA에 의한 이동 벡터 검출 방법은 하드웨어 관점에서 보면 이동 벡터 검출을 위한 sequential step이 문제이다. Ref. [6]의 "menu vector" search 방법은 temporal recursion을 도입하여 sequential step을 단 한번으로 줄이기는 하였으나, 이동 벡터 검출의 정확도 면에서 그 성능이 많이 떨어진다. 본 연구에서는 recursion을 도입한 "menu vector" search를 일반화하여 그 성능을 개선하였다.

(1). Recursive Block Matching 방법

입력 영상 신호를 sub-block으로 나눈 뒤, 이동 벡터의 관점에서 고려하기 쉽도록 motion vector field를 그림.2 와 같이 정의한다. 즉 현재의 block을 중심으로 수평방향을 k축, 수직 방향을 l축, 시간방향을 t축 으로 정하고 임의의 image sequence에 대한 각 sub-block의 이동 벡터를 motion vector field에서 표현하면, 현재 block의 이동 벡터는 $D(k, l, t)$ 로 표시할 수 있고, 이전의 field의 같은 위치의 이동 벡터는 $D(k, l, t-1)$ 과 같이 표현된다. Ref.[6]의 방법을 motion vector field를 사용하여 표현하면 다음 식과 같다.

$$D(k, l, t) = D(k, l, t-1) + U \text{ ----- (1)}$$

여기에서 U는 "menu vector" search에 의한 update term이다.

그러나 위의 방법은 frame사이에 변화가 심한 경우에는 성능이 크게 저하 된다. 이에 대한 해결책으로 식(1)을 변형하여 다음과 같이 생각할 수 있다. 이동 벡터의 initial estimate로서 과거의 여러개의 이동 벡터들을 고려하여 새로운 이동 벡터를 예측한 다음 "menu vector" search를 수행하여 보다 효과적으로 이동 벡터를 검출 할 수 있다. 즉 이동 벡터 검출에 관한 식을 linear prediction 기법을 적용하면 다음과 같이 일반화 하여 표현할 수 있다.

$$D(k, l, t) = [\sum_{i,j,n} D(k-i, l-j, t-n)] + U \text{ ----- (2)}$$

여기에서 $\sum_{i,j,n} = 1$ 이고, U는 "menu vector" search에 의한 update term 이다.

즉, linear prediction 기법을 적용하여 이동 벡터의 initial estimate를 구하고 "menu vector" search를 통하여 이동 벡터를 검출 할 수 있다.

4. 시뮬레이션 방법 및 결과

제 3장에서 제시한 방법의 성능을 비교하는 방법으로 그림.3 과 같은 시스템을 컴퓨터내에 구성하고 data base는 "Cronkite" 를 사용하여 시뮬레이션을 수행하였다. 이때의 "Cronkite" 영상은 크기가 256 * 256 이고 화소당 8비트로 샘플 되어 있으며 16 field를 갖는 연속 영상이다. 이 연속 영상의 첫번째 그림이 그림.7에 나타나 있다.

성능 비교의 방법으로 그림.3에서 sw1을 on, sw2를 off한 상태에서 예측 오차의 엔트로피를 측정하여 이동 벡터 검출의 정확도를 비교하고, sw1을 off하고 sw2를 on한 상태에서 NMSE(Normalized Mean Square Error)를 측정하여 성능을 비교 하였다.

여기에서 NMSE는 다음과 같다.

$$NMSE = 10 \log_{10} \frac{E(X_i - \hat{X}_i)^2}{E(X_i)^2}$$

시뮬레이션에서 사용된 block size는 8 * 8 이었고 최대 변위는 8pel 이며, 사용한 "menu vector"는 그림.4와 같다.

그림.5와 그림.6에 Ref.[6]에서 제안한 방법과 본 논문에서 제안한 방법중에서 $D(k, l, t) = (1/2) * [D(k-1, l, t) + D(k, l-1, t)] + U$ 인 방법을 예측 오차의 엔트로피와 NMSE 의 측면에서 나눴다.

5. 검토 및 결론

본 연구에서는 Motion-Compensated Coding 중에서 가장 핵심이 되는 문제인 이동 정보 검출 방법중에서 BMA에 대하여, teleconferencing이나 picture phone 영상의 경우, 이동 벡터의 spatial Correlation이 강하다는 사실을 이용하여, linear prediction 방법과 "menu vector" search 를 병용하여 이동 벡터를 검출 하므로써 search step을 단 한번으로 줄이고 성능도 향상 시켰음을 보였다.

그림.5의 결과를 보면 본 연구에서 제안된 방법은 기존의 "menu vector" search 방법 Ref.[6]에 비하여 예측 오차의 엔트로피 측면에서 0.25bit/pel 만큼 성능 개선이 되었음을 알 수 있고, NMSE 측면에서 평균적으로 3dB 만큼 개선 되었음을 보였다.

6. References

- [1] A. K. Jain, "Image Data Compression : A Review," Proceedings of the IEEE, Vol.69, No.3, March 1981
- [2] H. G. Mussman, P. Pirsh, H. G. Crallert, "Advances in Picture Coding," Proceedings of the IEEE, Vol.73, No.4, April 1985
- [3] J. R. Jain, A. K. Jaing, "Displacement Measurement and Its Application in Interframe Image Coding", IEEE Trans. on Commun., Vol.COM-29, No.12, Dec. 1981
- [4] T. Koga, et al, "Motion Compensated Interframe Coding for Video Conferencing, "National Telecommunication Conference, Nov.29 ~ Dec.3 1981
- [5] R. Srinivasan, K. R. Rao, "Predictive Coding on Efficient Motion Estimation," IEEE Trans. on Commun., Vol. COM-33, No.8, August 1985
- [6] Y. Ninomiya, Y. Ohtsuka, "A Motion Compensated Coding Scheme for Television Pictures," IEEE Trans. on Commun., Vol. COM-30, No.1, Jan. 1982

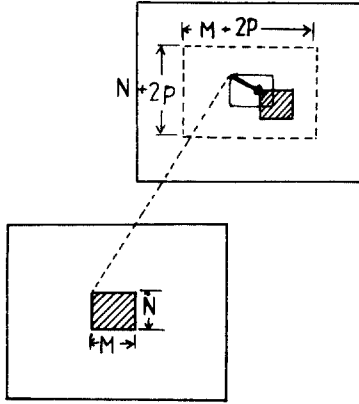


그림.1 BMA의 개요

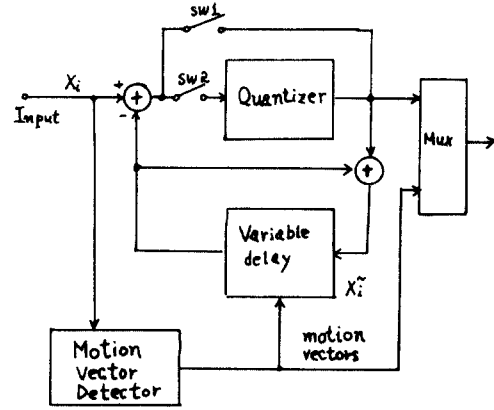


그림.3 Simulation에 사용한 시스템

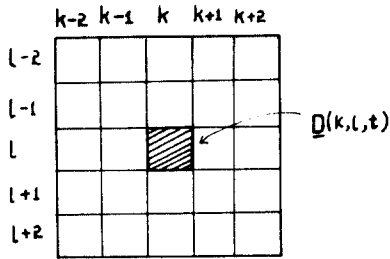


그림.2 Motion Vector Field의 정의

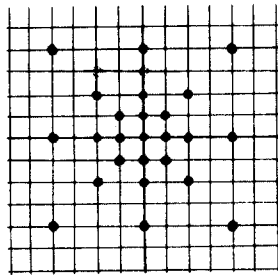


그림.4 Simulation에 사용한 menu vector set



그림.7 Simulation에 사용한 Data Base

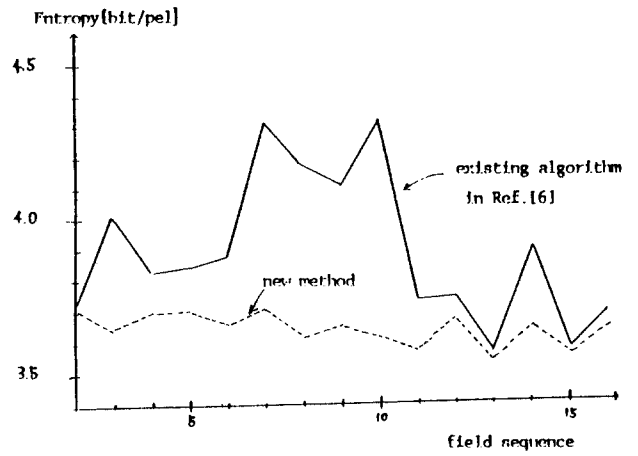


그림.5 예측 오차의 엔트로피

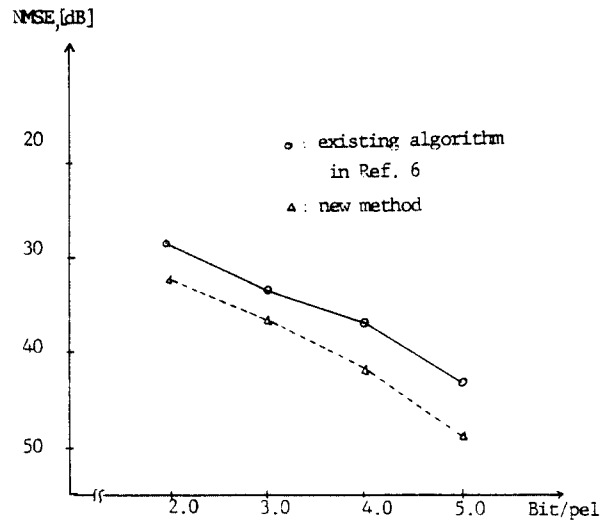


그림.6 전송 속도에 따른 NMSE