

탐색 영역 재설정을 이용한 고속 움직임 예측 방법

Fast Block Motion Estimation based on reduced search ranges in MPEG-4

김성제(Sungjei kim)*, 서동완(Dongwan Seo)**, 최윤식(Yoonsik Choe)***

Abstract - A block-based fast motion estimation algorithm is proposed in this paper to perform motion estimation based on the efficiently reduced search ranges in MPEG-4(ERS). This algorithm divides the search areas into several small search areas and the candidate small search area that has the lowest average of sum norm difference between current macroblock and candidate macroblock is chosen to perform block motion estimation using the Nobel Successive Elimination Algorithm (NSEA). Experimental results of the proposed algorithm show that the averaging PSNR improvement is better maximum 0.125 dB than other tested algorithms and bit saving effect is maximum 20kbps for some tested sequences in low-bit rate circumstance.

Key Words : Reduced Search Range, Fast Block Motion Estimation, MPEG-4.

1. 장 Introduction

최근 몇 년간 통신 기술의 급속한 발전으로 인해 이동 통신상에서의 멀티미디어 서비스에 대한 수요가 증가하고 있다. 하지만 비교적 적은 양의 데이터만을 전송해야하는 한계를 가진 이동 통신 채널 환경에서는 깨끗한 화질의 영상을 안정적이면서도 빠르게 서비스하는 일은 매우 어렵다. 동영상 부호화에서는 높은 데이터 압축률을 얻기 위해 시간적 중복성을 제거하는 움직임 보상 부호화가 널리 쓰이고 있으며, 이는 MPEG, H.263과 같은 국제 영상 부호화 표준에서 중요한 부분을 차지하고 있다. 움직임 보상 부호화 기법은 움직임 예측(Block Motion Estimation)을 통하여 현재의 매크로블록에 가장 유사한 매크로블록을 이전 프레임의 한정된 영역으로부터 취하고, 예측된 영상과 입력영상의 차를 변환 부호화하는 방법이다. 본 논문에서는 간단한 처리 과정과 데이터 접근의 일관성(regularity)을 유지하고, 이전 프레임의 한정된 영역인 탐색 영역(Search Range)을 단계적으로 줄여 실제적으로 움직임 예측 과정에서 많은 연산을 차지하고 있는 SAD(Sum of Absolute Differences) 연산을 줄이는 방법을 SEA(Successive Elimination Algorithm) [2-4]을 이용해서 MPEG-4 [1]에서 구현하였다.

2. 장 기존의 SEA 방법

2.1 절 SEA 알고리즘

BMA(Block Matching Algorithm)에서는 이전 프레임의

후보 매크로블록을 결정하기 위한 판정법으로 식 (1)과 같이 정의된 SAD를 사용한다.

$$SAD(x, y) = \sum_{i=1}^N \sum_{j=1}^N |f(i, j, t) - f(i+x, j+y, t-1)| \quad (1)$$

여기서 N 은 매크로블록의 크기이며, x, y 는 후보 매크로블록의 움직임 벡터를 의미하고 각각 움직임 벡터들은 움직임 탐색 영역에 의해 제한되어 있다. 그리고 $f(i, j, t)$ 는 해당 공간·시간적 위치 $((i, j), t)$ 의 화소값을 의미한다. SEA 알고리즘에서는 전역 탐색(Full Search)의 연산량을 줄이기 위해 다음과 같은 수학적 성질(식 (2))을 이용하여 탐색 후보 벡터의 개수를 줄이는 방법 [2]을 사용하였다.

$$SAD_{SN} = \left| \sum_{i=1}^N \sum_{j=1}^N |f(i, j, t) - \sum_{i=1}^N \sum_{j=1}^N |f(i+x, j+y, t-1)| \right| \leq SAD(x, y) \quad (2)$$

식 (2)에서 각각의 sum norm을 [2]에서 제안하는 고속 알고리즘으로 탐색 영역 내에서 계산한 뒤, 기준이 되는 움직임 벡터에서의 SAD를 계산하여 기준 SAD 값보다 작은 후보 움직임 벡터들에 대해서만 SAD 값을 계산하고 기준 SAD 값으로 갱신하여 탐색 영역 전체를 탐색하며 후보 탐색 영역을 줄이는 방법이다. 하지만 SAD_{SN} 은 $SAD(x, y)$ 의 값에 비해서 작은 문제점으로 인해 실제 SAD를 예측하는 판정 기준으로 사용하기에는 무리가 있다.

2.2 절 NSEA(Nobel SEA) 알고리즘 [3]

NSEA 알고리즘에서는 매크로블록을 4개의 하위 블록으로 나누어 각각 하위 블록의 합으로 새로운 SAD_{MSV} 을 제안하였는데, 식 (3)과 같다.

저자 소개

*연세大學 전기전자공학學科 碩士課程

**연세大學 전기전자공학學科 博士課程

***연세大學 전기전자공학學科 正教授·工博

$$SAD_{shSV} = \sum_{k=1}^1 \left| \sum_{i=1}^{N^2} \sum_{j=1}^{N^2} f(i,j,\theta) - \sum_{i=1}^{N^2} \sum_{j=1}^{N^2} f(i+x,j+y,t-1) \right| \quad (3)$$

$$SAD_{SN} \leq SAD_{shSV} \leq SAD \quad (4)$$

SAD_{shSV} 은 식 (4)와 같이 SAD_{SN} 보다 크거나 같기 때문에 [3] 본 논문에서 제안하는 ERS(탐색 영역 재설정) 알고리즘의 판정 기준으로 사용하였다.

3. 장 제안하는 탐색 영역 재설정 알고리즘

3.1 절 제안 동기

기존의 고속 알고리즘은 탐색 영역 내에서 움직임 벡터의 분포가 단일 형태(unimodal)라는 가정 하에 시작된 알고리즘이기 때문에 실제 영상에서 국소적인 최소치(local minimum)에 빠지기 쉬운 오류가 있고, 움직임이 큰 영상에서 전역 탐색 기법에 비해 현저하게 성능이 저하되는 문제점이 있다. 또한 하드웨어 디자인을 고려할 때 데이터 접근의 일관성이 떨어지는 단점이 있기 때문에 본 알고리즘을 제안한다.

3.2 절 탐색 영역 재설정 알고리즘

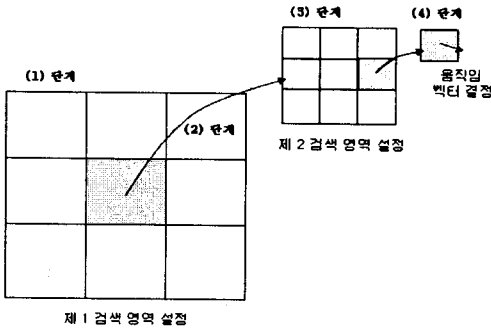


그림 1 탐색 영역 재설정을 통한 움직임 벡터 결정

(1) 초기 검색 영역을 9개의 사각형 검색 영역으로 동등하게 분할하여 제 1 검색 영역으로 설정될 수 있는 후보 검색 영역을 설정한다.

(2) 후보 검색 영역들 내의 검색 위치마다 참조프레임 내의 매크로블록 높과 현재 매크로블록 높의 차이의 절대값을 구하여 기준 $SAD(SAD_{ini})$ 보다 작아지는 해당 영역별로 모든 검색 위치 값들의 평균을 구하여 그 값이 가장 낮은 영역을 제 1 검색 영역으로 설정한다.

(3) 제 1 검색 영역을 다시 9개의 사각형 검색 영역으로 동등하게 분할하여 제 2 검색 영역으로 설정될 수 있는 후보 검색 영역을 설정하고 (2) 과정을 다시 반복하여 최종 단계 검색 영역을 결정한다.

(4) 최종적으로 결정된 검색 영역에 대해서 전역 탐색을 거쳐 SAD가 최소가 되는 위치에 해당하는 움직임 벡터를 구하여 움직임 예측을 수행한다.

3.3 절 기준 SAD(SAD_{ini})의 선택 방법

아래의 3가지 경우를 모두 계산한 뒤, 최소가 되는 SAD와

움직임 벡터를 결정하여 기준 SAD로 결정한다.

- (1) $MV(0,0)$ 에 해당하는 SAD
- (2) $MV(x,y) = \text{Med}(MV_A, MV_B, MV_C)$ 에 해당하는 SAD
- (3) SAD_{shSV} 를 최소화하는 위치에서의 SAD

식 (5)는 위 3가지 경우의 SAD에서 SAD_{ini} 선택 과정을 정리한 식이다.

$$\min_{SAD_{ini}} (SAD_{(1)}, SAD_{(2)}, SAD_{(3)}) \quad (5)$$

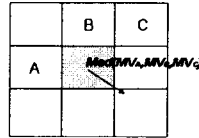


그림 2 기준 SAD (2)

0	1	0	1
1	0	1	0
0	1	0	1
1	0	1	0

그림 3 매크로블록의 탐색 패턴

3.4 절 Modified Algorithm [7]

SEA 알고리즘은 연산량이 비교적 많은 알고리즘이며 NSEA 알고리즘으로 sum norm을 계산해야하기 때문에 추가적으로 5.7%의 연산량 증가를 고려해야한다.[3] 따라서 연산량의 감소를 위해서 [7]에서 제안한 Modified Algorithm을 이용하여 그림 3과 같이 매크로블록의 탐색 패턴을 정한다.

'0'에 해당하는 매크로블록은 제안한 3.2, 3.3절의 탐색 영역 재설정 알고리즘을 이용하고, '1'에 해당하는 매크로블록은 3.3절의 (1), (2)에 해당하는 SAD 선택 방법과 최소값을 갖는 SAD의 움직임 벡터에서 수평 방향으로 ± 2 , 수직 방향으로 ± 1 의 탐색 영역을 두고 움직임 예측을 수행한 후 최종적인 움직임 벡터를 결정한다.

3.5 절 제안된 알고리즘의 매크로 블록 단위 연산량

(1) NSEA 알고리즘

매크로블록의 크기를 N^2 라고 했을 때, [2]에서 제안한 고속 알고리즘을 통한 sum norm의 계산으로 $4N^2$, sum norm 간의 차이 연산을 하는 과정에서 $16N^2$, 즉 $20N^2$ 의 연산이 필요하다.

(2) 기준 SAD 계산 및 최종 단계 전역 탐색

3개의 기준 SAD 계산과 최종 단계에서 평균적으로 4×3 전역 탐색을 하게 되므로 $15N^2$ 이 필요하다.

(3) 탐색 영역 재설정 과정

제 1 검색 영역과 제 2 검색 영역을 설정하는 과정에서 검색 영역별로 기준 SAD보다 작은 해당 위치들의 SAD_{shSV} 평균을 구해야하는 판정식이 필요하므로 평균적으로 $8N^2 \times 0.4 + 8N^2 \times 0.1 = 4N^2$ 의 연산이 필요하다.

(4) Modified Algorithm 적용

매크로 블록으로 서로 다른 탐색 패턴을 정의하므로 '0'에 대해서는 $39N^2$ 의 연산량이 필요하고, '1'에 대해서는 2개의 기준 SAD 결정 연산과 4×3 전역 탐색이 필요하므로 $14N^2$ 으로 제안한 알고리즘의 최종적인 연산량은 $(39+14)/2 = 26.5N^2$ 이 된다.

4. 장 실험 결과 및 결론

실험은 기준 비디오 영상인 foreman, news, coastguard, hall CIF(352×288) 영상을 사용하였고, 30 frames/s 입력 영상에 출력 영상은 10 frames/s에 Intra QP 5, Inter QP 16으로 두고 실험하였다. 탐색 영역은 [-16,15] 화소 단위에 매크로블록 단위는 16×16이었다. FS(전역 탐색, Full Search), TSS, CDHS 알고리즘으로 비교 실험하였고, 300 frames의 입력 영상을 사용하였다.

각각 알고리즘에 대해서 매크로블록별 연산량은 FS의 경우 탐색 영역 32×32에 대해서 1024N²이고, TSS의 경우 25N², CDHS는 탐색 영역이 영상에 따라서 다르기 때문에 4가지 영상에 대해서 탐색 위치들의 평균으로 표 1에 나타났다. 제안한 알고리즘은 26.5N²으로 3.5절에서 구하였다.

영상	Foreman	News	Coastguard	Hall
평균 탐색 개수	17.02	7.08	13.99	5.57

표 1 CDHS 평균 탐색 영역

CDHS의 경우 탐색 영역이 가변하기 때문에 하드웨어로의 디자인에 적합하지 않은 특성을 가지고 있으며 움직임이 큰 영상에 대해서 (표 2,4) 정확한 움직임 예측에 실패하고 있다. 또한 TSS는 움직임이 적은 영상에 대해서 제안한 알고리즘보다 많은 비트를 할당하는 문제점을 가지고 있다. (표 3,5) 대부분의 고속 알고리즘은 움직임이 많은 영상에 대해서 국소적인 최소치에 빠지게 되기 때문에 CDHS의 경우 움직임이 많은 영상에 대해서 평균 탐색량이 비교 알고리즘 중에 가장 빠르지만 화질의 열화와 비트율의 증가라는 문제가 발생한다.

제안하는 ERS 알고리즘은 데이터 접근의 일관성을 유지하고 있고, 움직임이 많고 적은 영상에 관계없이 평균적인 적용을 하고 있다. (표 2,3,4,5) foreman 영상의 경우 최대 20kbps의 비트절감효과와 0.16dB의 화질향상효과를 보이고 있다. 그림 4에서 카메라의 급격한 움직임이 있는 66 프레임부터 77 프레임까지 제안한 알고리즘이 비교 알고리즘보다 좋은 성능을 보이고 있다는 것을 알 수 있다.

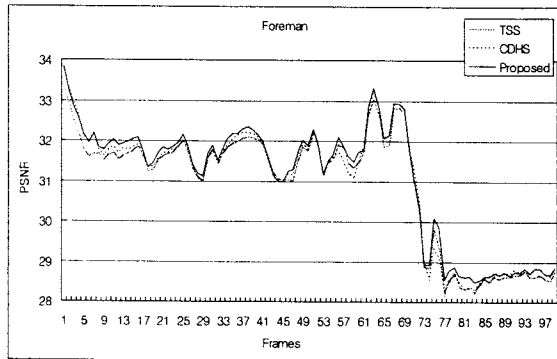


그림 4 각 프레임 별 Foreman 영상의 PSNR 비교

Foreman	FS	TSS	CDHS	ERS
PSNR(dB)	31.22	30.91	30.98	31.07
비트율(kbps)	143.18	177.85	167.73	152.72

표 2 Foreman 영상의 PSNR과 비트율

News	FS	TSS	CDHS	ERS
PSNR(dB)	33.41	33.36	33.29	33.32
비트율(kbps)	59.26	63.29	61.09	62.14

표 3 News 영상의 PSNR과 비트율

Coastguard	FS	TSS	CDHS	ERS
PSNR(dB)	29.00	28.91	28.69	28.86
비트율(kbps)	192.01	215.35	255.07	218.01

표 4 Coastguard 영상의 PSNR과 비트율

Hall	FS	TSS	CDHS	ERS
PSNR(dB)	34.18	34.17	34.13	34.16
비트율(kbps)	36.35	36.92	36.29	36.68

표 5 Hall 영상의 PSNR과 비트율

참 고 문 헌

- [1] ISO/IEC 14496-2 International Standard, Information technology coding of audio-visual objects: visual, 2000.
- [2] W. Li. and E. Salari, "Successive Elimination Algorithm for Motion Estimation", IEEE Trans, Image Proc., Vol.4, No.1, pp.105-107, Jan. 1995
- [3] S. Jung, S. shin, H. Baik and M. Park, "Nobel Successive Elimination Algorithms for the Estimation of Motion Vectors", IEEE Trans, Circuits Syst. Proc. vol. 2, pp. 616-619, Aug. 2000.
- [4] Y. Noguchi, J. Furukawa and H. Kiya, "A Fast Full Search Block Matching Algorithm for MPEG-4 Video", IEEE Image Processing, Proc. Int. conf. vol. 1, pp. 61-65, 1999.
- [5] T. Koga, K. Iinuma, T. Iijima, And T. Ishiguro, "Motion Compensated interframe coding for video conferencing", Proc. Nat. Telecommunications Nov. 1981, Conf., pp. G5.3.1-G.5.3.5
- [6] C. Zhu, X. Lin, L. P. Chau, K. P. Lim, H. A. Ang, and C.Y. Ong, "A novel hexagon-based search algorithm for fast block motion estimation", in Proc. IEEE Int. conf. Acoust., Speech, Signal Processing, May 2001.
- [7] H. S. Wang and R. M. Mersereau, "Fast Algorithms for the Estimation of Motion Vectors", IEEE Trans. Image Proc. vol. 8, no. 3, Mar. 1999.
- [8] C. H. Cheung and L. M. Po, "A novel cross-diamond search algorithm for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 12, no. 12, pp. 1168-1177, Dec. 2002.
- [9] C. H. Cheung and L. M. Po, "novel cross-diamond-hexagonal search algorithms for fast block motion estimation", IEEE Trans., Multimedia. vol. 7, no. 1, pp. 16-22, Feb. 2005.
- [10] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," IEEE Trans. Circuits Sys. Video Technol., vol. 3, pp.148-157, Apr. 1993.