

2단계 MAD 움직임 탐색을 이용한 MPEG 동영상 부호화

°이경환, 류권열
위덕대학교 멀티미디어공학과

MPEG Video Coding Using 2-Step MAD Motion Vector Search

°Kyeong-Hwan Lee, Gwon-Yeol Ryu
Dept. of Multimedia Engineering, Uiduk University

요 약

MPEG 동영상 부호화 중 BMA를 이용한 움직임 탐색과정에서, 왜곡적으로 MAD를 사용하는데 이는 블록 내 모든 화소에 동일하게 적용되므로 구획화 현상 등을 일으켜 주관적인 화질을 떨어뜨리는 원인이 된다. 본 논문에서는 블록 내 화소를 경계 영역과 내부 영역으로 나누어 경계 영역에 대한 MAD를 먼저 구하고, 문턱치를 비교하여 인터블록과 인트라블록으로 구분하고 인터로 분류된 블록에 대해서만 내부 영역의 MAD를 구하여 최종 보상블록을 결정함으로써 화질을 개선하고 계산량을 줄일 수 있었다.

1. 서 론

디지털 영상 신호는 높은 공간적 및 시간적 중복성(redundancies)을 가지므로, 이를 효과적으로 제거함으로써 높은 압축률을 얻을 수 있다. 동영상 부호화에서는 특히 인접 프레임(frame)이나 필드(field) 사이의 높은 시간적 중복성을 제거함으로써 정지 영상에서 보다 훨씬 높은 압축률을 얻을 수 있다. 시간적 중복성을 제거하는 대표적인 방법으로는, 화소 단위의 움직임 탐색(motion search) 및 보상(compensation) 방법인 PRA(pel recursive algorithm)와 블록 단위로 인접 프레임이나 필드에서 가장 왜곡이 작은 블록을 찾아서 추정 및 보상을 행하는 BMA(block matching algorithm) 방법 등이 있다[1]-[2].

멀티미디어에 대한 일반의 관심이 높아지고 이에 대한 기업들 간의 제품 호환성 문제가 대두됨으로 인해서, 동영상 압축 부호화에 대한 표준화가 시급하게 되었다. 따라서 동영상 표준화를 위하여 MPEG(Moving Picture Experts Group)이 구성되었고, 저장 매체용의 MPEG-1과 방송 및 통신용의 MPEG-2가 1992년과 1994년 각각 표준화 안으로 완성되었으며,

이를 기반으로 한 많은 응용 기술들이 연구되어 멀티미디어의 여러 분야에 적용되고 있다[3]-[4].

기본적인 MPEG 부호화 알고리즘에서는, PLA에 비해 계산량이 적은 BMA를 이용하여 프레임간의 시간적 중복성을 제거한다. 이때 블록들 간의 왜곡(distortion) 척도로 MSD(mean square difference)와 MAD(mean absolute difference)를 사용하는데, MSD는 객관적인 화질 측면에서 최적이지만 계산량이 많으므로 일반적으로 MAD를 사용하고 있다. 그러나 이들을 이용하여 블록 내의 전체 화소들에 대해 동일한 가중치로써 왜곡을 구하는 일반적인 BMA에서, 블록들 간의 닳음 정도를 나타내는 이들 값의 크기는 블록의 경계 영역의 화소들보다 내부 영역에 위치한 화소들에 의해 더 크게 좌우되므로 블록 경계 부분에서의 불연속이 생겨 구획화 현상(blocking artifact) 등 주관적인 화질의 열화가 일어난다[5].

본 논문에서는, 이러한 화질 열화를 줄이기 위해 블록 경계 영역 화소와 내부영역의 화소로 나누고, 경계 영역에 대한 MAD를 먼저 구하고 문턱치를 비교하여 인터블록과 인트라블록으로 구분하고 인터로 분류된

블록에 대해서만 내부 영역의 MAD를 구하여 최종 보상 블록을 결정하는 2단계 MAD 움직임 탐색 방법을 제안하였다. 모의 실험 결과, 제안한 방법은 기존의 방법에 비해 계산량을 줄이면서, 주관적인 화질은 화질을 개선할 수 있음을 확인하였다.

2. MPEG에서의 움직임 탐색 및 보상 방법

BMA에서는 참조 (reference) 프레임의 탐색 영역에서 블록 정합을 행하는 데, 화소 단위로 탐색점을 옮겨가며 현재 프레임의 매크로블록과 가장 왜곡이 작은 블록의 위치를 움직임 벡터로 구한다. 이때 현재 매크로블록 b 의 움직임 벡터 $V(b)$ 는,

$$V(b) = [d_h, d_v]^T \quad (1)$$

와 같이 표현되고, 여기서 d_h 및 d_v 는 각각 움직임 벡터의 요소들로 수평 및 수직 방향의 변위 (displacement)를 나타낸다. 그림 1에서는 BMA에서의 움직임 벡터를 나타내었다. MPEG의 P 및 B-프레임에서는, BMA를 이용하여 현재 매크로블록의 움직임 벡터를 구하고 그 특성에 따라 인터 또는 인트라로 분류한다. 즉, 현재 블록과 움직임 보상된 블록과의 차 블록의 분산과 현재 블록의 분산을 비교하여, 차 블록의 분산이 문턱치 (threshold)보다 작거나 현재 블록의 분산보다 작을 경우에만 현재 블록은 인터로 선택된다. 그렇지 않은 경우에는 현재 블록을 인트라로 선택한다. 이는 블록의 움직임이 탐색 영역을 벗어날 경우 또는 탐색 대상이 없어진 경우에는 움직임 보상 부호화의 효율이 떨어지기 때문에 이를 방지하기 위함이다. 인터블록은 움직임벡터와 움직임 보상된 차 성분의 DCT (discrete cosine transform) 계수가 부호화되고, 인트라블록의 경우 현재 블록을 직접 DCT 부호화하여 전송된다.

MPEG에서 움직임 탐색시 현재 블록과 참조 프레임의 탐색 블록간의 왜곡 계산법 (distortion measure)으로 MSD와 MAD가 사용된다. 블록의 크기를 $S \times S$ 라 하면, 두 블록 b_1 과 b_2 사이의 MSD는

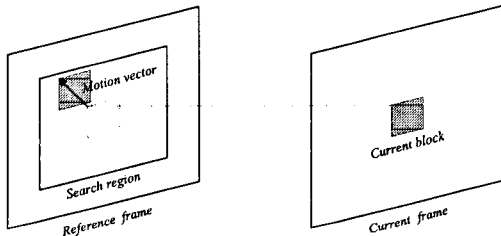


그림 1. BMA 방법에서의 움직임 벡터

$$MSD(b_1, b_2) = \frac{1}{S^2} \sum_{x=0}^{S-1} \sum_{y=0}^{S-1} |b_1(x,y) - b_2(x,y)|^2 \quad (2)$$

와 같이 계산된다. 이때, x, y 는 블록 내 화소의 좌표이고, $b(x,y)$ 는 그 위치의 화소의 밝기 값이다. MSD는 탐색 영역의 후보 블록 중 현재 블록과 벡터적인 거리 상 최적 최소 (optimal minimum)의 왜곡을 가지는 블록을 찾을 수 있지만, 매 번의 탐색점마다 각각의 화소들 간에 제곱연산을 행해야 하므로 많은 계산 시간이 필요하게 된다. 따라서 대부분의 MPEG 부호화기에서는 MSD 대신 계산량을 줄이고자 MAD를 사용하고 있다. 동일한 조건에서의 MAD는

$$MAD(b_1, b_2) = \frac{1}{S^2} \sum_{x=0}^{S-1} \sum_{y=0}^{S-1} |b_1(x,y) - b_2(x,y)| \quad (3)$$

와 같이 계산된다.

3. 제안한 2단계 움직임 탐색 방법

본 논문에서는 기존의 MPEG 부호화기에서와 같이 움직임 보상을 행하였을 때, 보상 블록의 블록 경계 영역의 화소의 왜곡이 심해지는 단점을 개선하기 위하여 2단계 움직임 탐색 (two-step motion search; TSMS) 방법을 제안하였다.

그림 2에서와 같이 매크로블록 내의 화소를 경계 영역과 내부 영역의 두 부분으로 나누었다. 첫번째 단계에서는 현재 블록과 탐색 영역의 후보 블록 사이에 블록 경계 영역의 화소만을 이용하여 MAD를 구하였다. 이를 BMAD라고 하고

$$BMAD(b_1, b_2) = \frac{1}{S_{RB}} \sum_{(x,y) \in R_b} |b_1(x,y) - b_2(x,y)| \quad (4)$$

와 같이 계산된다. 이때, R_B 는 블록 내에서 BMAD 영역을, S_{RB} 는 그 영역의 화소 수를 나타낸다. 그리고 BMAD가 정한 문턱치보다 작은 후보 블록들만이 다음 단계로 넘어가게 하였다.

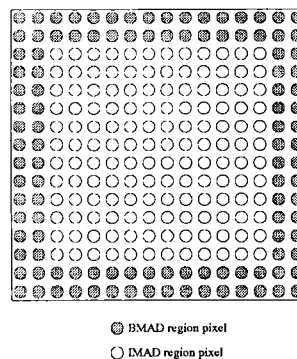


그림 2. 제안한 TSMS 방법에 사용할 블록 경계 영역 및 내부 영역의 화소

기존의 MPEG에서 원 블록과 보상 블록의 분산을 구하여 인터 및 인트라블록의 분류를 하던 것과 달리, 제안한 방법에서는 모든 탐색점의 BMAD 값이 문턱치보다 크면 인트라로 분류하여 그에 따른 계산량을 줄였다. 한편 문턱치는 인터블록과 인트라블록의 비율이 MPEG에서와 동일하여 발생하는 비트가 같게 유지되도록 통계적으로 정하였다.

두번째 단계에서는 BMAD 값이 문턱치를 넘지 않는 후보 블록들에 대해 현재 블록과 블록 내부 영역의 화소들만을 이용하여 MAD를 구하는 IMAD 탐색을 행하는데 이는

$$IMAD(b_1, b_2) = \frac{1}{S_{R_i}} \sum_{(x,y) \in R_i} |b_1(x,y) - b_2(x,y)| \quad (5)$$

와 같이 계산된다. 이때, R_i 는 블록 내에서 IMAD 영역을, S_{R_i} 는 그 영역의 화소 수를 나타낸다.

제안한 TSMS 방법에서의 최종 MAD는 BMAD와 IMAD의 합으로

$$MAD(b_1, b_2) = BMAD(b_1, b_2) + IMAD(b_1, b_2) \quad (6)$$

와 같이 구하였으며, 이 값이 최소가 되는 후보 블록을 보상 블록으로 정한다.

4. 제안한 중첩 2단계 움직임 탐색방법

본 논문에서는 또한 움직임 탐색을 행할 때에 인접 블록들의 화소들을 포함시킨 중첩 2단계 움직임 탐색(overlapped TSMS; OTSMS) 방법을 제안하였다. 앞에서 제안한 TSMS 방법은 첫 번째 단계에서 계산된 BMAD가 작은 후보 블록에 대해서만 두 번째 단계에서 선별적인 탐색을 행하므로 화소 오차 계산을 행할 때 계산량이 대폭 줄게 된다. 따라서 제안한 OTSMS 방법에서는, MPEG에서 보다 계산량의 증가없이 중첩 탐색을 할 수 있다.

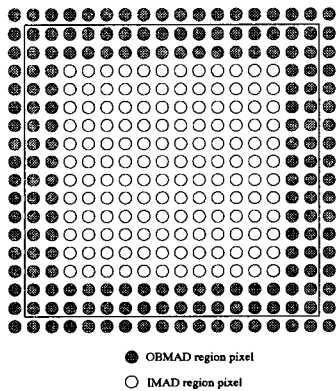


그림 3. 제안한 OTSMS 방법에 사용할 블록 경계 영역 및 내부 영역의 화소

움직임 탐색을 위한 첫번째 단계에서는, 그림 3에서와 같이 블록 내의 블록 경계 영역뿐만 아니라 블록 경계 너머의 이웃 블록과 중첩된 화소들을 이용하여 현재 블록과 탐색 영역의 후보 블록간의 MAD를 구하는 OBMAD 탐색을 행하였다. OBMAD는

$$OBMAD(b_1, b_2) = \frac{1}{S_{R_{OB}}} \sum_{(x,y) \in R_{OB}} |b_1(x,y) - b_2(x,y)| \quad (7)$$

와 같이 구하였다. 이때, R_{OB} 는 블록 내에서 BMAD 영역을, $S_{R_{OB}}$ 는 그 영역의 화소 수를 나타낸다. 이때 인터 및 인트라 블록의 분류는 BMAD 방법에서와 동일하게 하고, 전체 MAD 또한

$$MAD(b_1, b_2) = OBMAD(b_1, b_2) + IMAD(b_1, b_2) \quad (8)$$

로 하여 이 값이 최소인 후보 블록을 보상 블록으로 정하였다.

5. 모의 실험 결과 및 결론

제안한 방법 각각의 성능을 평가하기 위해 컴퓨터 모의 실험을 행하였다. 실험 영상으로는 352×240 SIF(source input format)의 움직임이 많은 FOOTBALL, 카메라 움직임이 있는 FLOWER GARDEN 각 50 프레임, 화상 전화 영상인 SALESMAN 50 프레임, 그리고 720×480 DTV 영상인 MOBILE 50 프레임을 사용하였다. 이때 움직임 보상은 순방향, 완전복원 방법을 사용하였으며, 인터 및 인트라 블록을 분류하는 문턱치로 BMAD는 20, OBMAD는 25를 사용하였다.

표 1에서는 일반적인 MPEG을 기준으로 한 제안한 방법의 평균 계산량 증감을 나타내고 있는데, 1단계 탐색 후 문턱치를 비교하여 2단계에서 선별적으로 탐색을 행하므로 TSMS방법은 계산량이 30% 이상 줄어들 수 있고, OTSMS 방법은 블록 경계 너머의 중첩된 화소들을 탐색하였는데도 계산량이 늘어나지 않음을 볼 수 있다.

그림 4에서는 일반적인 MPEG에서와 같이 BMA로 움직임 보상을 하였을 때, 인터블록의 보상블록과 원 블록 간의 평균 자승 화소 오차를 나타내고 있다. 블록 경계 영역의 화소들에 비해서 블록 내부 영역 화소들이 더 많은 수의 유사 화소들을 가지므로 MAD 값의 크기는 블록의 경계 영역의 화소들보다 내부 영역에 위치한 화소들에 의해 더 크게 좌우되어 블록 경계 부분에서의 왜곡이 심해짐을 볼 수 있다. 그러므로 DCT 등 손실 부호화로 차영상을 부호화할 경우 본원 영상에서 구획화 현상이 나타나게 된다.

그림 5 및 6에서는 제안한 TSMS 및 OTSMS방법에 의해 구해진 인터블록의 보상 블록 평균 자승 화소

오차를 나타내고 있는데, 그림 4의 결과와 비교할 때 제안한 방법으로 블록 경계 영역의 화소 오차를 많이 줄일 수 있음을 볼 수 있다. 특히 OTSMS 방법은 블록 경계 너머의 중첩된 화소들을 이용함으로써 블록 경계 지점 화소의 왜곡이 두드러지는 현상이 거의 제거됨을 볼 수 있다.

따라서 제안한 방법을 실제 MPEG 코덱에 적용하였을 경우, 움직임 탐색에 필요한 계산량을 줄일 수 있고 보상블록과의 왜곡이 블록 경계영역에 집중되는 단점을 극복할 수 있어 복원 영상의 구획화 현상 등의 주관적인 화질 열화를 줄일 수 있을 것이다.

[참고문헌]

[1] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," *Proceeding of IEEE*, vol. 83, no. 6, June, 1995.

[2] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," *Proceeding on National Telecommunication Conference*, pp. G5.3.1-G5.3.5, Nov. 1981.

[3] K. R. Rao and J. J. Hwang, *Techniques and Standards for Image, Video, and Audio Coding*: Prentice Hall, 1996.

[4] J. L. Mitchell, W. B. Pennebaker, C. Fogg, and D. J. LeGall, *MPEG Video Compression Standard*: Chapman and Hall, 1997.

[5] S. A. Karunasekera and N. G. Kingsbury, "A Distortion Measure for Blocking Artifacts in Images Based on Human Visual Sensitivity," *IEEE transactions on Image Processing*, pp. 713-724, Vol. 4, No. 6, Jun. 1995.

표 1. 제안한 방법의 계산량 증감

Proposed methods	The increment of operations per block		The increasing ratio [%]	
	Additions	Multiplications	Additions	Multiplications
TSMS	-73368	+1	-33.47	0
OTSMS	-16780	+1	-7.66	0

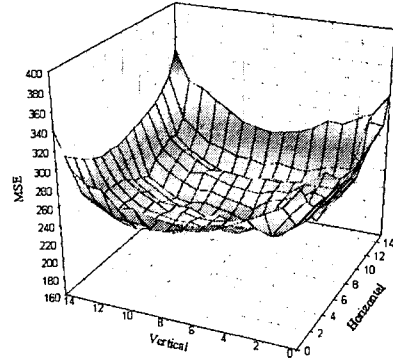


그림 4. 일반적인 MPEG에서와 같이 움직임 보상을 했을 경우의 인터블록 평균 자승 화소 오차

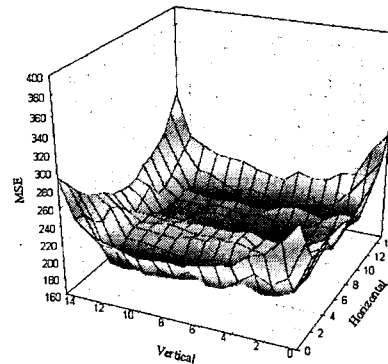


그림 5. 제안한 TSMS 방법으로 보상을 했을 경우의 인터블록 평균 자승 화소 오차

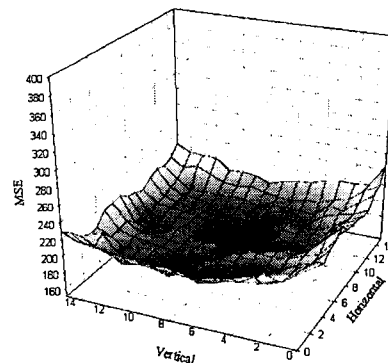


그림 6. 제안한 OTSMS 방법으로 보상을 했을 경우의 인터블록 평균 자승 화소 오차