가변 블록 움직임 보상을 위한 개선된 고속 모드 결정법

이제윤, 최웅일, 전병우
성균관대학교 정보통신 공학부

Enhanced Fast Mode Decision for Variable Block Motion Compensation

Jeyun Lee, Woongil Choi, Byeungwoo Jeon
Dept. of Information and Computer Engineering, Sungkyunkwan University
E-mail : jeyun22@ece.skku.ac.kr

Abstract

최근 표준화가 완성된 H.264 는 가변 블록 움직임 보상, 복수 참조 영상, 그리고 1/4 화소 움직임 벡터 정확도를 지원하고 있다. 그러나 이러한 새로운 부호화 기술은 부호화 효율 향상의 주된 요인이면서, 동시에 높은 복잡도의 요인이기도 하다. 따라서 H.264 비디오 표준의 실제 응용 확대를 위해서는 이러한 기술의 속도 향상이 필수적이다. [1]에서 제안한 고속 모드 결정법은 조기에 모드 결정을 할 수 있기 때문에, 움직임 벡터 탐색과 비트율-왜곡치 (Rate-Distortion cost) 계산 과정을 효율적으로 생략할 수 있는 방법이다. 하지만 [1]에서 제안된 측정치 r 은 주변 블록의 정보를 이용하지 않기 때문에 모드 결정 에러를 좀 더 효과적으로 줄이지 못했다. 본 논문에서는 주변 블록의 정보를 이용하여 [1]의 방법을 개선시킨 것으로 실험 결과 큰 부호화 손실 없이 계산량 감소에 있어 매우 높은 효율을 제공함을 확인하였다.

I. 서론

H.264 의 주요 부호화 기술 중 가변블록 움직임 보상 기술은, 기존의 H.263 이나 MPEG-2, 4 보다 움직임 보상 블록의 크기를 더욱 세밀하게 나눔으로서 영상의 특징이나 영상내 움직임 특성에 잘 부합하는 기술이다. 그림 1 과 같이 움직임 보상의 크기는 16x16 에서 4x4 까지 있으며, 8x8SUB 는 8x8 블록 단위로 움직임 보상 구조를 가질 수 있는 모드이다. 즉 8x8, 8x4, 4x8, 4x4 중 한가지 모드를 8x8 블록 단위로 가질 수 있으며, 각 8x8 에 대한 모드 정보를 복호기에 전송해 주어야 한다.

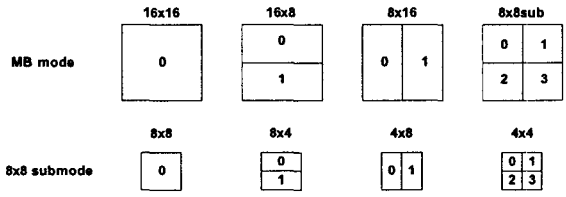

그림 1 움직임 보상을 위한 매크로블록 분할

그림 1 에서 비교적 평활한 영역이나 물체의 크기가 큰 경우에는 16x16 처럼 큰 블록으로 움직임이 보상될 확률이 높지만, 복잡하고 물체의 크기의 작은 영상은 4x4 처럼 작은 블록으로 움직임이 보상이 될 확률이 높다. 그리고 하나의 매크로블록은 여러 크기의 블록으로 나뉘어질 수 있으며, 각 세부 블록들에 대해 움직임 벡터 탐색이 이루어 지고, 움직임 벡터에 대한 정보도 각 블록마다 전송해 주어야 한다. 따라서 큰 블록이 선택된다면 전송해야 하는 움직임 벡터의 비트량이 적지만, 이 때 발생되는 잉여 데이터의 비트량이 많게 된다. 반대로 작은 블록으로 세분화 되어 움직임 보상이 이루어진다면 잉여 데이터의 비트량은 작지만, 움직임 벡터가 많이 발생되어, 움직임 벡터의 비트량이 증가하게 된다. 따라서 어떤 크기로 움직임 보상을 할 것인가를 결정하는 것은 압축률과 밀접하게 연관되어 있으며, 이

를 효율적으로 결정하기 위하여 H.264 는 비트율-왜곡 최적화 기법 (Rate-Distortion Optimization method) 을 도입하였다[2].

복수 참조 영상 기법은 과거 여러 개의 참조 영상 중 현재 블록과 유사도가 가장 높은 것을 찾아 이를 움직임 보상에 사용하는 것으로, 그림 1 의 각 움직임 보상 블록들은 후보 참조 영상 마다 움직임 벡터 탐색을 수행하고 이중 잉여 데이터가 가장 적은 참조 영상을 결정한다. 이러한 과정은 참조 영상을 하나만 이용하는 기존의 H.263 이나 MPEG-2, 4 보다 많은 계산량을 필요로 하며, 전체 복잡도에서 움직임 보상이 80% 이상 차지하는 주된 요인이다[3]. 그림 1 에서 블록 크기가 8x8 이상인 블록들은, 각각의 분할마다 참조 영상을 다르게 가질 수 있다. 즉, 4 개의 8x8 블록들은 서로 다른 참조 영상을 가질 수 있는 것이다. 복수 참조 영상 기법을 이용하여 부호화 할 때, tempet 나 mobile & calendar 영상처럼 반복적인 움직임을 갖고 있는 영상인 경우, PSNR 이 최고 1dB 까지 향상 되었다[4]. 참조 영상수에 대한 권고 사항은 프로파일과 레벨에 맞춰 명시되어 있다[5].

H.264 는 움직임 벡터의 정확도를 1/4 화소 단위까지 지원하고 있다. 최초, 정수 위치의 화소들을 {1, -5, 20, 20, -5, 1}/32 의 계수를 갖는 6-tap 필터를 이용하여 1/2 위치의 화소들을 보간한다. 다음으로 정수 위치 화소와 1/2 위치의 화소들을 선형 보간하여 1/4 위치의 화소들을 생성한다. 이렇게 1/4 위치의 화소를 생성하는 과정은 모든 참조 영상마다 수행되어 지는 것으로, 바로 이전 영상만을 참조 영상으로 사용했던 기존의 비디오 표준보다 많은 계산량을 필요로한다.

H.264 의 주요 부호화 기술 중 비트율-왜곡 최적화 기법은 가변 블록 움직임 보상, 움직임 벡터 선택, 참조 영상 결정등에 쓰이는 기술로서, 여러 가능성 중에서 부호화기의 효율을 극대화 하는 것을 결정하는 방법이다. 이 기술은 Lagrangian 최적화 기법을 기반으로 하여 고안된 [2]의 부호화 기법을 이용한 것으로, 비트율-왜곡 최적화 기법을 사용하여 10%의 비트 감소가 있음이 보고 되었다[6].

위에서 설명한 가변 블록 움직임 보상, 복수 참조 영상, 1/4 화소 정확도를 갖는 움직임 벡터, 비트율-왜곡 최적화 기법등으로, H.264 는 MPEG-4 SP (Simple Profile) 보다 동일한 비트율에서는 PSNR 이 2dB 높으며, 동일한 PSNR 에서는 압축률이 두 배나 좋다[7][8]. 하지만 높은 부호화 효율을 제공하는 이러한 기술들은 높은 복잡도의 주요 요인으로서, MPEG-4 SP 보다 14~16 배 복잡도가 높다[6].

II. 고속 모드 결정법

H.264 의 매크로 블록 모드는 SKIP, MB16X16, MB16X8, MB8X16, SUB8X8, INT4, INT16 이 있다. SKIP 모드는 움직임 보상 블록 크기가 16x16 이고, 움직임 벡터가 (0, 0) 이거나, PMV (Predicted Motion Vector)와 같으며, 잉여 데이터에 대한 변환 계수가 모두 0 인 모드이다. 그리고 INT4, INT16 은 각각 4x4, 16x16 단위의 공간 예측 모드를 의미한다. 9 가지 모드 중에서 최적 모드를 결정하기 위해 비트율-왜곡 최적화 기법을 사용하는데 이러한 과정은 많은 계산량을 필요로 한다. 그래서 [1]은 부호화 효율 감소를 최소화 하면서 계산량을 효율적으로 줄일 수 있는 방법을 제시하였다.

[1]에서 제안한 고속 모드 결정법은 인터 매크로블록 모드를 Class16={SKIP, MB16x16, MB16x8, MB8x16}, Class8={SUB8x8}로 나누고, 각 Class 의 대표 모드인 MB16x16 과 MB8x8 의 비트율-왜곡치인 RDcost 를 구하여, 작은 RDcost 를 갖는 모드의 Class 에서 최적 모드가 결정되되도록 하는 방법이다. 하지만 이런 흑백 논리적인 모드 결정법은 잘못된 모드로 결정될 확률이 높기 때문에, 이를 해결하고자 식 (1)과 같은 측정치 r 을 정의 하였다. 여기서 RDcost8 과 RDcost16 은 MB8x8, MB16x16 의 비트율-왜곡치이다.

$$r = \frac{RDcost8 - RDcost16}{RDcost16} \times 100 \qquad (1)$$

그리고 r 에 따른 모드 결정 에러 확률 분포 곡선을 구하여, 모드 결정 에러가 많이 발생하는 부분은 기존 방법 대로 부호화 하고, 에러가 상대적으로 적은 부분에서는 제안한 방법을 적용하였다. 즉, r 이 'th_low' 보다 작다면 Class8 에서 최적 모드가 결정되도록 하고, 'th_high' 보다 크다면 Class16 에서 최적 모드가 결정되도록 하며, 'th_low'와 'th_high' 사이에서는 기존 방법대로 모드가 결정되도록 하는 것이다. 여기서 'th_low'와 'th_high'는 r 에 따른 모드 결정 에러 확률 분포 곡선의 표준편차인 'std'를 이용하여 구한 것으로 식 (2)와 같다.

$$th\_low = mean - \frac{std}{2},\ th\_high = mean + \frac{std}{2} \qquad (2)$$

인접한 매크로블록들의 정보에는 상관도가 매우 높

기 때문에, 움직임 벡터를 부호화할 때 주변 블록의 움직임 벡터를 이용하고, H.263 의 공간 예측 부호화 기술도 주변 블록의 변환 부호화 계수 값들을 이용하는 등, 주변 블록의 정보를 이용한 많은 부호화 기술들이 소개되어져 왔다. 따라서, [1]에서 제안한 측정치 r 도 주변 블록의 정보를 효율적으로 이용한다면, 모드 결정 에러를 효과적으로 줄일 수 있을 것이다.

표 1 은 왼쪽과 윗쪽 매크로블록 Class 가 결정되었을 때, 현재 매크로블록의 Class 발생 빈도를 100 을 기준으로 하여 나타낸 것 이다. 그리고 freq16, freq8 은 Class16 과 Class8 의 발생 빈도를 나타낸다. 예를 들어 왼쪽 블록이 Class16 이고 윗쪽 블록이 Class8 일 때 현재 매크로블록에서 Class16 이 100 번 중에 64.64 번 정도 발생한다는 의미이다. 표 1 은 foreman, new, container, silent 영상으로부터 산출한 통계적인 데이터이다. Boundary 는 픽쳐의 외곽 부분을 의미하고, ClassINT 는 공간 예측 모드에 대한 Class 를 의미한다.

표 1. Class 발생 빈도

Above / Left	Boundary		Class16		Class8		ClassINT	
	freq16	freq8	freq16	freq8	freq16	freq8	freq16	freq8
Boundary	81.82	18.18	95.84	4.16	39.10	60.90	62.50	37.50
Class16	95.64	4.36	90.66	9.34	64.64	35.36	70.00	30.00
Class8	60.95	39.05	70.62	29.38	45.54	54.46	46.91	53.09
ClassINT	85.19	14.81	73.64	26.36	52.12	47.88	62.70	37.30

표 1 의 발생 빈도와 측정치 r 을 효과적으로 결합하는 것이 중요하다. 즉, Class16 의 발생 빈도가 높다면 측정치 r 도 쉽게 Class16 이 선택 될 수 있도록 바꿔어야 할 것이다. 이를 위하여 MB16x16, MB8x8 의 비트율-왜곡치인 RDcost16, RDcost8 을 식 (3)과 같이 변환하였다. 그리고 RDcost8′ , RDcost16′ 을 이용하여, 식 (1)의 측정치 r 을 계산한다.

$$RDcost8' = RDcost8 + \frac{freq16}{freq8 + freq16} \times 100$$
$$RDcost16' = RDcost16 + \frac{freq8}{freq8 + freq16} \times 100 \quad (3)$$

식 (3)에서 Class16 의 발생 빈도인 freq16 이 큰 값이라면, RDcost8′ 의 값이 커지게 되어 현재 블록의 모드가 Class16 으로 선택될 가능성이 커지게 되고, 반대로 freq8 이 크다면, 현재 블록의 모드가 Class8 로 선택될 가능성이 커지게 된다. 따라서 [1]에서 사용되었던 임계치 'th_low', 'th_high' 를 사용하지 않더라도, 측정치 r 의 부호에 따라 Class 를 분류할 수 있게 되었다. 즉 r 이 음수이면, Class8 에서 최적 모드가 결정되도록 하고, r 이 양수라면, Class16 에서 최적 모드가 결정되도록 하는 것이다. 그리고 표 1 의 Class 발생 빈도수는 프레임이 부호화 될 때 마다 갱신하여 영상의 특징에 적응 할 수 있도록 한다.

III. 실험 결과

실험 조건은 H.264 표준화 그룹에서 권고하고 있는 공통 실험 조건 [9]을 따라 표 2 와 같이 하였다. 실험에 사용된 부호화기는 JM(Joint Model) 6.1d 이며, 실험은 베이스라인 프로파일에 맞추어 수행하였다. 그리고 [1]에서는 공간 예측 부호화 과정을 생략하는 방법도 제시하였는데, 본 실험에서는 이 기법을 그대로 적용하였다.

표 2. 실험 조건

	news (qcif)	container (qcif)	foreman (qcif)	silent (qcif)	paris (cif)	mobile (cif)	tempete (cif)
Frame rate (Hz)	10	10	10	15	15	30	30
Total frames	100	100	100	150	150	300	250
QP	28, 32, 36, 40						
Coding option used	R-D optimization, Hadamard transform IPPP structure, CAVLC, do not use any error tools						
codec	JM 6.1d encoder						

표 3 은 개선된 고속 모드 결정법에 대한 결과이다. 이때 부호화 성능을 평가하기 위한 객관적 지표로, BDBR (Bjonteggard Delta BitRate)과 BDPSNR (Bjonteggard Delta PSNR) [10]을 이용하여 나타내었다. 이들은 각각 비교하고자 하는 두 방법 간의 비트율과 PSNR 차이의 평균을 의미한다. 그리고 BDBR 의 (+) 부호와 BDPSNR 의 (-) 부호는 부호화 성능의 손실을 나타낸다. 그리고 $R_{inter}, R_{intra}, R_{total}$ 은 줄어든 비트율-왜곡치 계산 수에 대한 백분율을 나타낸 것이며, 이에 대한 수식은 다음과 같다.

$$R_{mode} = \frac{\#RDcost_{mode}\{REF\} - \#RDcost_{mode}\{proposed\}}{\#RDcost_{mode}\{REF\}} \times 100 \quad (4)$$

식 (4)에서 '#RDcost'는 비트율-왜곡치 계산 회수를 나타내고, 'REF'는 기존의 JM6.1d 를 의미하며, 'proposed'는 제안된 방법을 의미한다. 표 3 에서 감소한 인터 모드의 비트율-왜곡치 계산 수는 37.5%, 공간 예

측 모드인 경우 56.56%의 계산량 감소가 있었다. 그래서 전체적으로 줄어든 비트율-왜곡치 계산 수는 49.34% 이다. 반면 부호화 손실은 BDBR 이 약 4%, BDPSNR 이 0.19dB 정도로 미비하다. 개선된 고속 모드 결정법은 [1]의 고속 모드 결정법에 비해, BDBR 은 0.93%, BSPSNR 은 0.04dB 정도의 부호화 손실이 더 발생하였지만, 비트율-왜곡치 계산 회수에 있어서 인터 모드의 계산 수가 [1]의 결과 보다도 11.33% 더 줄어들어, 부호화 이득의 큰 손실 없이 [1]의 고속 모드 결정법 보다, 더욱 많은 비트율-왜곡치 계산 수 감소를 얻을 수 있다.

표 3. 개선된 고속 모드 결정법에 대한 결과

	BDBR [%]	BDPSNR [dB]	R_{inter} [%]	R_{intra} [%]	R_{total} [%]
container	4.65	-0.23	37.50	66.82	54.30
foreman	5.08	-0.27	37.50	59.01	49.82
news	3.20	-0.18	37.50	70.31	56.29
mobile	3.32	-0.14	37.50	14.36	24.18
silent	4.40	-0.21	37.50	68.50	55.24
paris	3.37	-0.17	37.50	79.14	67.89
tempete	3.53	-0.14	37.50	37.76	37.65
average	**3.94**	**-0.19**	**37.50**	**56.56**	**49.34**
Results of [1]	**3.01**	**-0.15**	**26.17**	**57.03**	**45.05**

III. 결론

H.264 에서 가변 블록 움직임 보상 구조와 비트율-왜곡 최적화 기법은 부호화기 전체 복잡도의 대부분을 차지하고 있는데, 본 논문에서 제안한 개선된 고속 모드 결정법은, 모드 결정 초기에 Class 를 분류하여 모드 결정 과정을 간략화하기 때문에, 비트율-왜곡치 계산 과정과 움직임 벡터 탐색 과정을 생략할 수 있었다. 제안된 방법은 [1]의 고속 모드 결정법을 개선한 것으로, 인터 모드 결정시 사용되었던 임계치 'th_low', 'th_hight' 대신 주변 블록들의 Class 정보를 이용하여, 모드 결정 정확도를 높일 수 있는 개선된 고속 모드 결정법을 제안하였다. 그리고 제안된 방법은 매 프레임이 부호화 될 때 마다 표 1 의 Class 빈도 수를 갱신하기 때문에 영상의 움직임 특성에 적응적인 고속 모드 결정이 가능하다.

개선된 고속 모드 결정법은 간단한 연산으로 얻은 측정치 r 을 이용하고도, 큰 부호화 손실 없이 약 50% 정도의 비트율-왜곡치 계산 수 감소를 얻을 수 있었다. 그리고 [1]의 고속 모드 결정법과 비등한 부호화 효율을 가지면서도 더욱 많은 비트율-왜곡치 계산 수 감소를 달성할 수 있었다. 따라서 제안된 방법은 부호화기 전체 복잡도를 현저하게 낮추어 고속 부호화기의 구현을 가능하게 하였다.

참고문헌

[1] W. Choi, J. Lee, S. Yang, and B. Jeon, "Fast motion estimation and mode decision with variable motion block sizes," VCIP, Lugano, July. 2003.

[2] G. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression," Draft for submission to IEEE Signal Proc. Magazine, Vol.15 pp74-90 (1998).

[3] C. Blanch and K. Denolf, "Memory Complexity Analysis of the AVC Codec JM1.7," ISO/IEC JTC1/DC29/WG11 MPEG02/M8378, Fairfax, May 2002.

[4] W. Choi and B. Jeon, "Selective fast motion estimation with variable motion block sizes," International Workshop on Advanced Image Technology (IWAIT'03), pp. 101-106, Nagasaki, 21-22, Jan. 2003.

[5] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)," Doc. JVT-G050r1, May. 2003.

[6] M. Zhou, "Evaluation and Simplification of H.26L Baseline Coding Tools," Doc. JVT-B030, Jan, 2002.

[7] P. Topiwala, G. Sullivan, A. Joch and F. Kossentini, "Overview and Performance Evaluation of the Draft ITU-T H.26L Video Coding Standard," Proc. SPIE, Appl. Dig. Im. Proc, Aug. 2001.

[8] ITU-T SG16 Q6, "Performance Evaluation of H.26L, TML 8 vs. H.263++ and MPEG-4," Doc. VCEG-N18, Sep. 2001.

[9] G. Sullivan and G. Bjontegaard, "Recommended simulation common conditions for H.26L coding efficiency experiments on low-resolution progressive-scan source material," ITU-T Q.6/16, Doc. VCEG-N81, Sep. 2001.

[10] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," ITU-T Q.6/16, Doc. VCEG-M33, Mar. 2001.