

영상압축 효율 향상을 위한 움직임 벡터 부호화 방법

*송관용, 최광표, 주영훈,
삼성전자

e-mail : {kwsong71, kp5.choi, mrzoo}@samsung.com,

정봉수, 전병우
성균관대학교 정보통신공학부

e-mail : bsjung@ece.skku.ac.kr, bjeon@skku.edu

Motion Vector Coding for Improved Video Coding Efficiency

Kwanwoong Song, Kwang Pyo Choi, Younghun Joo
Samsung Electronics

Bongsoo Jung, Byeungwoo Jeon
School of Information and Communication Engineering
Sungkyunkwan University

Abstract

We propose a new motion vector skip coding method for better motion compensation-based coding of inter-slices in H.264/AVC. It is to best utilize the spatial correlation between motion vectors of adjacent 4x4 blocks by effective motion vector coding. For this purpose, we introduce a new macroblock type of Predictive (P) slice into those of the H.264/AVC, so that it can lead to reduction in the coding bits required for encoding motion information. Experimental results with several well-known test video sequences verify that better performance of the proposed method is obtained.

I. 서론

H.264/AVC는 ITU-T VCEG(Video Coding Experts Group)과 ISO/IEC MPEG(Moving Picture Experts Group)이 공동으로 최근 표준화 완료한 비디오 압축 표준으로, 종래 비디오 압축표준들과 같이 블록기반

부호화방식을 유지하면서도 우수한 비트율-왜곡 (RD: Rate-Distortion) 성능을 보여준다[2]. 특히, 높은 부호화 효율과 다양한 전송환경에 대한 유연성을 제공하기 때문에 H.264/AVC는 모바일 영상통신, 무선네트워크를 통한 비디오 스트리밍 서비스 및 고화질 디지털방송(HDTV: High Definition Television)등에 널리 사용되고 있다. H.264/AVC는 기술적으로 공간 방향 인트라 예측, 가변 블록 움직임 보상, 다중 참조 프레임, in-loop 디블록킹 필터, CAVLC(Context Adaptive VLC), 그리고 CABAC(Context-based Adaptive Binary Arithmetic Coding)과 같은 새로운 부호화 기술들을 사용하고 있다. 또한, 그림 1에서 보는 것처럼 움직임 보상은 16x16에서부터 4x4에 이르기까지 다양한 크기의 가변 블록들에 의해 수행되며, H.264/AVC는 매크로블록 부호화를 위해 SKIP, P16x16, P16x8, P8x16, P8x8 등의 인터모드와 4x4블록크기 인트라 예측의 I4MB와 16x16블록크기 인트라 예측의 I16MB, 그리고 High Profile에서 추가로 제공되는 8x8블록크기 인트라 예측의 I8MB의 3가지 인트라모드를 지원한다. P8x8 인터모드는 각각의 8x8블록 안에서 또다시 8x8, 8x4, 4x8, 4x4중의 하나로 선택되어질 수 있는 계층적 구조를 가진다. 즉, 종래 H.263 및 MPEG-4 Part-2 표준

본 연구결과는 2007년 11월 31일에 종료된 삼성전자 산학과의 연구결과물임.

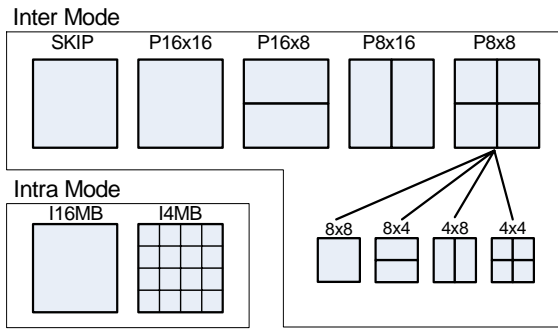


그림 1. P슬라이스의 매크로블록 모드

과는 달리 더욱 세분화된 움직임보상 블록 크기를 사용하여 매우 세밀한 움직임 예측을 하기 때문에 더욱 높은 압축 효율을 얻을 수 있음을 알 수 있다. 그러나, 이러한 다양한 가변 블록크기 움직임 보상 기술을 사용함에 따라 하나의 매크로블록에서 발생할 수 있는 움직임벡터는 최대 16개가 된다. 이것은 움직임벡터 부호화에 많은 비트량이 소요됨을 보여준다.

일반적으로 움직임 보상 블록의 크기가 작아질수록 부호화할 잔차(residual) 데이터가 적게 발생하는 것은 장점이지만, 작은 블록 크기 단위로 움직임 보상을 할 경우, 각 블록마다 발생하는 움직임 벡터들을 효과적으로 부호화 하지 않을 경우 오히려 움직임 벡터수의 증가에 따른 비트량 증가라는 역효과가 더 커질 수 있다. 이 경우 작은 블록들을 이용한 움직임 보상을 제대로 살릴 수 없게 된다. 움직임 벡터는 주변 블록의 중간값 예측 움직임 벡터(pmv: predicted motion vector)와 상관도가 매우 높기 때문에 H.264/AVC는 움직임 벡터 부호화를 위해 현재 블록의 움직임 벡터와 예측 움직임 벡터의 차분 값을 부호화 한다. 또한 매크로블록에서 각각의 블록들의 움직임 벡터는 서로 독립적으로 부호화되며, 예측 움직임 벡터와 동일한 경우라도 최소 2비트가 소요된다. 이것은 매크로블록 내의 모든 블록들이 각각의 예측움직임 벡터와 동일한 경우에도 서로 독립적으로 부호화되어 움직임 벡터 부호화 효율 감소를 발생시킨다.

따라서 본 논문에서는 이러한 문제를 해결하고자 4x4 움직임 보상 블록의 움직임 벡터들을 모두 전송하지 않아도 되는 생략기반 움직임벡터 부호화 방법을 제시한다.

본 논문의 구성은 다음과 같다. II장에서는 H.264/AVC의 움직임 벡터 부호화와 모드 결정법을 기술 하고, III장에서는 제안하는 움직임 벡터 생략 부호화 방법을 설명하며, IV장에서는 제안 알고리즘의 성능을 보이고, 마지막으로 결론을 맺는다.

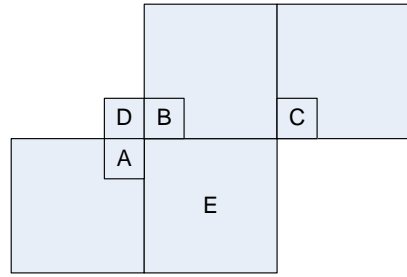


그림 2. 움직임 벡터 예측

II. H.264/AVC 움직임 벡터 부호화 및 매크로블록 부호화모드

H.264/AVC는 각 블록을 부호화함에 있어 움직임벡터의 부호화 효율을 높이기 위해 그 움직임벡터를 그대로 부호화하는 것이 아니라 인접 블록들의 움직임벡터들을 이용하여 예측 부호화 한다. 그림 2에는 H.264/AVC 표준에 따른 예측 움직임벡터를 구하기 위한 주변 블록들이 나타나 있다. H.264/AVC에서 움직임 벡터 부호화는 이웃한 주변블록들(A, B, C)의 3개의 움직임벡터의 중간값과 차분한 데이터를 부호화한다. 차분 움직임 벡터 (*mvd*: motion vector difference)는 다음의 식 (1)과 같이 계산된다.

$$mvd = mv_E - pmv \tag{1}$$

$$pmv = median(mv_A, mv_B, mv_C)$$

여기서 mv_E 는 부호화할 현재 블록 E의 움직임 벡터이고 예측 움직임 벡터 pmv 는 그림 2에서처럼 주변블록 A, B, C의 움직임 벡터, mv_A, mv_B, mv_C 의 중간값으로 산출된다. 블록 C가 슬라이스 경계 밖에 있는 경우와 같이 mv_C 가 유효하지 않은 경우에는 식 (1)에서 mv_C 를 mv_D 로 대체하여 계산한다. 움직임 벡터는 주변블록의 움직임 벡터들과 상관도가 매우 높기 때문에 차분 움직임 벡터 mvd 는 확률적으로 영벡터가 될 가능성이 높다.

H.264/AVC는 mvd 가 0인 경우, 움직임 벡터의 x 축 1비트와 y 축 1비트가 필요하여 블록당 2비트로 부호화된다. 결과적으로, 만약 4x4움직임 보상블록에서 16개의 4x4블록 모두가 예측 움직임 벡터 pmv 와 같은 경우에도 최소 32비트의 움직임 벡터 정보가 필요하게 된다.

일반적으로 움직임이 적은 배경이나 동일한 움직임을 갖는 큰 물체의 경우, 움직임벡터는 그 주변 블록들의 움직임벡터와 같을 확률이 매우 크다. 이러한 사실을

영상압축에 활용하기 위해 H.264는 P(Predictive) 슬라이스 부호화시 SKIP 모드를 매크로블록 모드중의 하나로 정의하여 사용한다. 이 SKIP모드는 해당 매크로블록이 각각 다음을 만족하는 것을 의미한다.

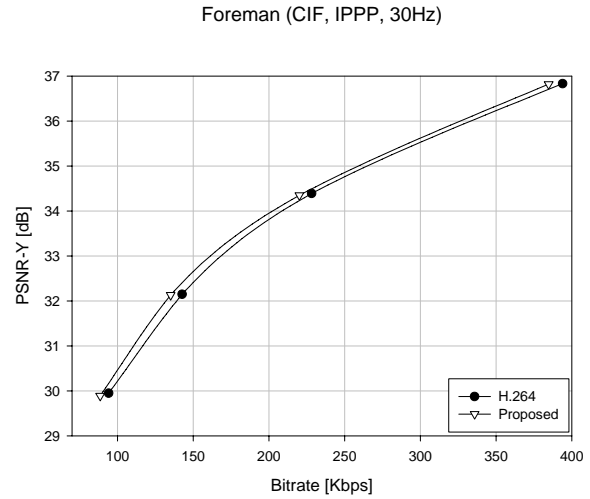
- 매크로블록의 움직임 보상 블록 크기 = 16X16
- $mvd=0$: 움직임벡터가 그 주변 블록들로부터 유도해 낸 예측 움직임 벡터와 같음
- CBP(Coded Block Pattern)=0 : 움직임 보상 후 잉여데이터의 부호화 비트량이 발생하지 않음

이 조건들을 모두 만족하는 매크로블록에 대하여 부호화기는 움직임벡터와 잉여데이터 정보를 부호화하는 대신, SKIP모드라는 것을 매크로블록 부호화모드 정보로만 표현하도록 하여 상당한 비트율 감소를 달성하였다[1]. 한편, 양방향예측을 사용하는 B 픽취 (또는 슬라이스)의 경우도 이와 유사한 다이렉트(direct) 모드가 매크로블록 모드로 존재한다. 이 다이렉트 모드에서도 역시 움직임벡터에 대한 정보를 전송하지 않는다. 대신, 복호화기는 인접 슬라이스에 있는 동일 위치의 블록이 갖는 움직임벡터로부터 양방향의 움직임벡터 두 개를 유도하게 된다. 이러한 다이렉트 모드도 역시 움직임벡터정보를 전송할 필요가 없기 때문에 비트율을 상당히 감소시킬 수 있다.

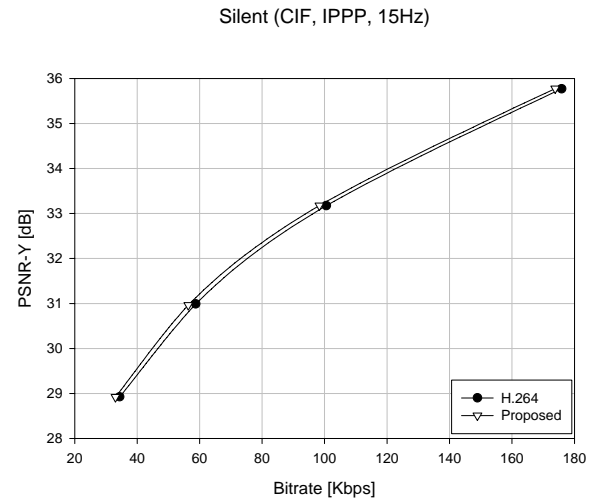
따라서 H.264/AVC의 참조 모델인 JM(Joint Model) 부호화기는 각각의 블록크기에 대하여 움직임 탐색을 한 다음 움직임 보상 블록들의 부호화 효율을 율-왜곡(Rate-Distortion) 최적화 기법으로 계산하고, 부호화 효율이 가장 높은 것을 최적 움직임 보상 블록으로 결정한다.

III. 움직임 벡터 생략 부호화 방법

부호화 블록의 움직임 벡터는 주변블록의 중간값 예측 움직임 벡터와 상관도가 매우 높다. H.264/AVC는 4개의 4x4블록 모두가 각각의 예측 움직임 벡터와 동일한 경우에도, 각 8x8블록에 대하여 항상 8비트가 필요하다. 또한, 움직임 보상 블록의 크기가 작을수록 잔차(residual signal) 신호가 작음에도 불구하고 16개의 4x4 블록 각각이 예측 움직임 벡터와 동일한 경우에도 항상 32비트 움직임 벡터 부호화 데이터가 필요하다. 따라서 제안하는 움직임 예측 방법은 residual 블록의 에너지가 적게 발생하는 4x4블록 움직임 보상을 수행하면서 움직임 벡터를 전송하지 않도록 움직임 벡터를 생략 부호화한다. 이를 위하여 기존 H.264/AVC의 매크로블록 타입에 이러한 SkipMV 모드를 나타내는 새로운 mb_type 을 정의 하였다. 이런 이유로 제안 방법은 H.264/AVC와 더 이상 호환되지 않는다.



(a) Foreman



(b) Silent

그림 3. 비트율-왜곡 성능의 비교

제안하는 움직임 벡터 생략 부호화 (SkipMV) 모드는 움직임 보상을 4x4 블록으로 하며, 4x4블록들의 움직임 벡터가 예측 움직임 벡터 pmv 와 동일함과 참조영상은 모두 바로 이전 픽취(refIDX=0)임을 의미한다. 제안하는 움직임 벡터 생략 모드를 나타내기 위하여 5비트의 VLC code로 부호화되는 skipMV라는 mb_type 을 정의하였다. 만약 SkipMV 모드로 선택된 경우, H.264/AVC는 P8x8을 나타내는 모드정보 5비트, 움직임 벡터 32비트, 그리고 P8x8 서브블록의 타입정보(4개의 8x8 서브블록 각각에 대한 4x4 움직임 보상 모드 정보 5비트)에 대하여 20비트로, 총 57비트로 부호화된다. 하지만 제안 방법은 단지 매크로블록 모드 정보

에 대한 오직 5비트로 부호화가 가능하다. 제안한 움직임 벡터 생략 부호화 모드는 *SkipMV* 모드의 움직임 벡터 비용을 계산하기 위해 추가적인 움직임 탐색과정이 불필요하기 때문에 추가적인 복잡도 증가도 크게 발생하지 않는 장점이 있다.

IV. 실험 결과

본 논문에서 새로히 소개하는 움직임 벡터 생략 기법의 성능을 검증하기 위하여 이를 JM12.3 참조소프트웨어[5]에 구현하였다. 객관적 성능 평가를 위해 H.264/AVC의 부호화 매크로블록 모드(Anchor)와 제안하는 부호화 매크로블록 모드(Proposed)의 성능비교를 하였다. 실험 조건은 다양한 영상, QP, 그리고 부호화 프레임율에 대하여 다음과 같이 설정하였다.

- 1) 저복잡도 모드(Low Complexity Mode) 사용
 - 1) 고속 움직임 탐색 사용: EPZS[6]
 - 2) 탐색 영역: +/-16
 - 3) 하다마드 변환 부호화 사용
 - 4) CAVLC 사용
 - 5) IPPP 구조

사용한 영상은 Foreman(CIF, 30Hz), Coastguard(CIF, 30Hz), Paris(CIF, 15Hz), Silent(CIF, 15Hz)이며, QP ∈ {28, 32, 36, 40}으로 하여 실험 하였다.

또한, 제안한 매크로블록 모드의 성능의 평가에는 BDBR(%)과 BDPSNR(dB)을 사용하였다[3]. 여기서 BDBR(%)의 (-)와 BDPSNR(dB)의 (+) 부호는 제안 부호화방법의 성능향상을 나타낸다.

표 1의 결과로부터 제안된 움직임 벡터 생략 기법의 적용으로 BDBR을 평균 2.31% 감소시킬 수 있음을 볼 수 있다. 이것은 제안한 움직임 벡터 생략 기법을 사용하여, 4x4블록의 작은 블록으로 움직임 보상을 하면서도 움직임 벡터 부호화에 들어가는 상당한 정보의 전송 없이 움직임 벡터 부호화가 가능하기 때문이다.

그림 3은 Foreman과 Silent 영상의 비트율-왜곡 성능 그래프로 나타낸 것이다. 그림 3에서 비트율이 낮아질수록 제안 방법의 부호화 성능이 더욱 좋아 지는 것을 볼 수 있다. 이것은 움직임 벡터와 매크로블록 모드 정보가 저 비트율로 갈수록 전체 부호화 비트에서 차지하는 비중이 커지기 때문이다.

V. 결론

본 논문은 H.264/AVC의 부호화 효율 향상을 위한 움직임 벡터 생략 부호화 방법을 제안하였다. 움직임 벡터 생략 부호화 방법은 16개의 4x4 움직임 보상 블록

표 1. 제안 부호화기의 성능 비교

Sequence	BDPSNR[dB]	BDBR[%]
Foreman	0.18	-3.70
Coastguard	0.06	-1.65
Paris	0.06	-1.19
Silent	0.11	-2.71
Average	0.10	-2.31

의 움직임 벡터를 전송하지 않고 이웃블록의 예측 움직임 벡터를 직접 사용하는 방법으로 H.264/AVC의 인터 매크로블록 타입에 움직임 벡터 생략 모드를 추가하여 동작하는 것이다. 제안 방법은 H.264/AVC와 비교하여 평균 2.31% 부호화 비트를 감소시켰다.

참고문헌

- [1] 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, Mar. 2003.
- [2] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, G. J. Sullivan, "Rate-Constrained Coder Control and Comparison of Video Coding Standards," IEEE Trans. on Circuits Syst. Video Technol., vol. 13, no. 7, pp.688-703, Jul. 2003.
- [3] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," VCEG Contribution VCEG-M33, Austin, Apr. 2001.
- [4] X. Lu, A. M. Tourapis, and J. Boyce, "Fast Mode Decision and Motion Estimation for JVT/H.264," Proc. of IEEE Conference on Image Processing (ICIP), pp.853-856, 2003.
- [5] <http://bs.hhi.de/~suehring/tml/download/jm12.3.zip>
- [6] X. Lu, A. M. Tourapis, and J. Boyce, "Fast Mode Decision and Motion Estimation for JVT/H.264," Proc. of IEEE Conference on Image Processing (ICIP), pp.853-856, 2003.