

일반논문-10-15-3-11

Phased-in 코드를 이용한 움직임 벡터 예측기의 효율적인 부호화 방법

문지희^{a)}, 최정아^{a)}, 호요성^{a)‡}

Efficient Coding of Motion Vector Predictor using Phased-in Code

Ji-Hee Moon^{a)}, Jung-Ah Choi^{a)}, and Yo-Sung Ho^{a)‡}

요 약

H.264/AVC 비디오 압축 표준은 압축 효율을 높이기 위해 다양한 크기의 블록을 사용하여 화면 사이의 움직임 예측을 수행한다. H.264/AVC는 가변적인 블록 크기의 움직임 보상을 통해 세밀한 영역의 움직임까지 예측할 수 있어 잔여 영상을 나타내는 정보량을 효과적으로 줄일 수 있다. 복호를 위해서는 각 블록의 움직임 벡터를 전송해야 하는데, 저비트율 환경에서는 움직임 벡터 정보가 전체 비트스트림의 약 40%를 차지한다. 움직임 벡터 정보량을 줄이기 위해 비디오 부호화 전문가 그룹(VCEG)에서는 다양한 움직임 벡터 예측(Motion Vector Competition) 방법을 제안하였다. 다양한 예측 움직임 벡터를 사용하여 실제 전송해야 할 움직임 벡터 차분값(Motion Vector Difference, MVD)의 크기를 줄이기 때문에 압축 효율을 높일 수 있다. 그러나 다양한 예측 움직임 벡터를 사용하기 때문에 선택된 예측 움직임 벡터의 인덱스 정보를 복호기로 전송해야 한다. 이 논문에서는 인덱스 정보를 효율적으로 전송하기 위해 Phased-in 코드를 기반으로 한 새로운 코드워드 표를 제안했다. 실험을 통해 제안한 방법을 이용하여 동일한 화질에서 평균 약 7.24%의 비트율을 절감할 수 있었고, 동일한 비트율에서는 평균 약 0.36dB의 화질을 향상시킬 수 있었다.

Abstract

The H.264/AVC video coding standard performs inter prediction using variable block sizes to improve coding efficiency. Since we predict not only the motion of homogeneous regions but also the motion of non-homogeneous regions accurately using variable block sizes, we can reduce residual information effectively. However, each motion vector should be transmitted to the decoder. In low bit rate environments, motion vector information takes approximately 40% of the total bitstream. Thus, motion vector competition was proposed to reduce the amount of motion vector information. Since the size of the motion vector difference is reduced by motion vector competition, it requires only a small number of bits for motion vector information. However, we need to send the corresponding index of the best motion vector predictor for decoding. In this paper, we propose a new codeword table based on the phased-in code to encode the index of motion vector predictor efficiently. Experimental results show that the proposed algorithm reduces the average bit rate by 7.24% for similar PSNR values, and it improves the average image quality by 0.36dB at similar bit rates.

Keywords : motion vector competition, KTA, motion vector difference, motion vector predictor

I. 서 론

a) 광주과학기술원 정보통신공학과

Gwangju Institute of Science and Technology

‡ 교신저자 : 호요성 (hoyo@gist.ac.kr)

· 접수일(2010년3월26일), 수정일(2010년4월27일), 게재확정일(2010년5월17일)

H.264/AVC는 압축률이 우수한 비디오 부호화 표준으로 연합 비디오 팀(Joint Video Team, JVT)에 의해 2003년 5월 제정되었다. 그러나 H.264/AVC는 개발 당시 모바일

TV 환경에서 VCD급 품질을 목적으로 개발되어 HD급 영상에서는 높은 성능을 보이지 못한다. HD 콘텐츠에 익숙해진 사용자들이 점차 고화질 서비스를 요구하게 되면서 사용자의 수요를 충족시키기 위해 현재 H.264/AVC보다 높은 압축 성능을 위한 차세대 비디오 부호화 표준화가 진행되고 있다.

H.264/AVC 비디오 표준화 이후 VCEG에서는 압축 효율 개선, 복잡도 개선, 피드백 기반 에러 강인성을 핵심 기술 분야(Key Technology Area, KTA)로 선정하여 차세대 비디오 압축에 대한 기술을 연구했다. 최근 고화질 비디오 부호화(High Efficiency Video Coding, HEVC) 표준화를 진행하기로 결정함에 따라 VCEG KTA 기술의 중요성이 높아졌다. KTA에서 제안한 여러 기술 중에서 비트율-왜곡 최적화 기반의 다양한 움직임 벡터 예측(Motion Vector Competition, MVC)은 움직임 벡터 정보를 보다 효율적으로 부호화하는 방법이다.

H.264/AVC는 가변적인 블록 크기 단위로 움직임 보상을 수행하여 압축 효율을 높일 수 있지만, 저비트율 환경일 경우 전체 비트스트림에서 움직임 벡터의 비율이 약 40%를 차지한다^[1]. 움직임 벡터 정보량을 줄이기 위해 KTA에서는 다양한 예측 움직임 벡터 후보군을 사용하여 움직임 벡터와 예측 움직임 벡터의 차분 값을 줄이는 방법을 제안했다. 실제 움직임 벡터를 부호화하는데 사용하는 비트를 줄일 수 있지만, 선택된 예측 움직임 벡터의 인덱스를 복호기로 전송해야 한다는 단점이 있다.

이 논문의 2장에서는 기존의 움직임 벡터 부호화와 KTA에서 제안한 다양한 움직임 벡터 예측 방법을 기술하고, 3장에서 제안하는 예측 움직임 벡터 인덱스 부호화를 설명한다. 4장에서는 제안한 방법의 성능을 비교하기 위해 실험한 결과를 보이고, 마지막으로 5장에서 결론을 맺는다.

II. 움직임 벡터 예측 방법

1. 움직임 벡터 부호화

H.264/AVC 표준의 움직임 예측 및 보상 기술은 부호화

해야 할 매크로블록과 유사한 예측 매크로블록을 결정한 뒤, 원본 매크로블록과 예측 매크로블록의 잔여 영상을 전송한다. 부호기에서는 예측 매크로블록과 전송된 잔여 영상을 더하여 복호를 수행하므로 부호기에서 결정된 예측 매크로블록의 위치 정보를 나타내는 움직임 벡터를 전송해야 한다.

움직임 벡터의 정보를 나타내기 위해서 차동 펄스 부호 변조(Differential Pulse Code Modulation, DPCM)방법이 사용된다. 한 화면에서 위치하는 인접한 매크로블록을 이용하여 현재 부호하고자 하는 움직임 벡터를 예측하면 실제 움직임 예측을 통해 결정된 움직임 벡터와 유사하다. 그러므로 움직임 벡터의 원본 값보다는 차분 값(Motion Vector Difference, MVD)의 크기가 더 작다. 식 (1)은 움직임 벡터와 예측 움직임 벡터의 차분 값을 나타낸다. mv 는 움직임 예측을 통해 결정된 움직임 벡터이고 pmv 는 인접한 매크로블록을 통해 예측된 움직임 벡터를 나타낸다.

$$MVD = mv - pmv \tag{1}$$

예측 움직임 벡터는 인접하는 매크로블록들의 움직임 벡터들의 중간 값으로 결정된다. 그림 1은 현재 블록 E의 움직임 벡터를 예측하기 위해 미디언 예측을 사용했을 때의 참조 블록의 위치를 보여준다.

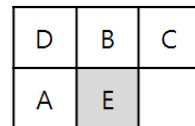


그림 1. 움직임 벡터 예측을 위한 참조 블록
Fig. 1. Current and neighboring macroblocks

현재 블록 E의 예측 움직임 벡터는 이미 부호화된 주변의 블록(A, B, C)의 움직임 벡터를 이용하여 미디언 예측에 의해 얻어진다. 이를 식으로 나타내면 식 (2)와 같다.

$$pmv = median(mv_A, mv_B, mv_C) \tag{2}$$

현재 블록이 영상의 경계 부분에 위치하면 주변에 블록이 없는 경우가 생긴다. 이럴 경우에는 H.264/AVC에서 정

의한 방법에 따라 예측 움직임 벡터를 결정한다. 예를 들어, A 블록이 없는 경우에는 (0, 0)으로 A 블록의 움직임 벡터를 대체한다. 그리고 C 블록이 없는 경우에는 D 블록의 움직임 벡터를 대체하여 사용하며, B 블록과 C 블록이 없는 경우에는 A 블록의 움직임 벡터를 사용한다^[2].

복호기에서도 복호된 주변의 블록을 이용하여 예측 움직임 벡터를 계산한 후, 전송된 움직임 벡터의 차분 값을 더하여 현재 블록의 움직임 벡터를 복호한다. 벡터의 차분 값은 부호가 있는 Exponential Golomb(signed Exp-Golomb)코드를 이용하여 부호한다^[3].

2. 다양한 움직임 벡터 예측 방법

KTA에 구현되어 있는 다양한 움직임 벡터 예측 방법은 움직임 벡터의 차분 값 크기를 줄여 압축 효율을 향상시킨다. 기존의 방법은 예측 움직임 벡터를 미디언 예측기만을 이용하여 결정했지만, 다양한 움직임 벡터 예측 방법은 다수의 예측 움직임 벡터 후보군을 생성한다. 이 중 최소의 비트를 사용하여 움직임 벡터 차분 값을 부호하는 후보 움직임 벡터가 현재 부호화하고자 하는 매크로블록의 예측 움직임 벡터로 결정된다. 그림 2는 예측 움직임 벡터 후보군을 나타낸다.

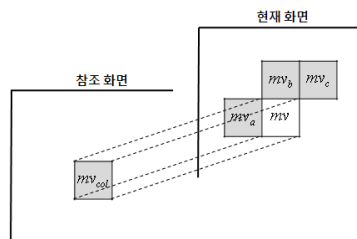


그림 2. 예측 움직임 벡터의 후보군
Fig. 2. Candidate motion vector predictor set

공간적인 특성을 반영한 예측 움직임 벡터의 후보로는 이미 부호화된 인접한 주변 블록의 움직임 벡터(mv_a, mv_b, mv_c)와 H.264/AVC 미디언 예측기에 의해 결정된 벡터(mvmedian)가 있다. 시간적 특성을 반영한 예측 움직임 벡터의 후보로는 참조 화면에서 현재 부호화하고자 하는 매크로블록과 동일한 위치에 있는 블록의 움직임 벡터(mvcollocated)가 있다. 추가적인 예측 움직임 벡터 후보로는 (0, 0)이 있다. SKIP 모드의 경우, 위에서 설명한 예측 움직임 벡터 후보들과 함께 부가적으로 확대된 공간적 예측 움직임 벡터 후보(extended spatial predictive motion vector)를 사용한다. 확대된 공간적 예측 움직임 벡터 후보는 미디언 예측기에서 결정된 벡터(mvmedian)와 유사하지만 이미 부호화된 주변 블록의 이용 여부에 의해 결정되는 점이 다르다. A, B, C 블록을 모두 이용할 수 있으면 확대된 공간적 예측 움직임 벡터 후보는 미디언 예측에 의해 결정된다. 그러나 세 개의 인접한 블록 중 한 개만 이용 가능하면 해당 매크로블록의 움직임 벡터로 결정된다^[4].

크로블록과 동일한 위치에 있는 블록의 움직임 벡터(mvcollocated)가 있다. 추가적인 예측 움직임 벡터 후보로는 (0, 0)이 있다. SKIP 모드의 경우, 위에서 설명한 예측 움직임 벡터 후보들과 함께 부가적으로 확대된 공간적 예측 움직임 벡터 후보(extended spatial predictive motion vector)를 사용한다. 확대된 공간적 예측 움직임 벡터 후보는 미디언 예측기에서 결정된 벡터(mvmedian)와 유사하지만 이미 부호화된 주변 블록의 이용 여부에 의해 결정되는 점이 다르다. A, B, C 블록을 모두 이용할 수 있으면 확대된 공간적 예측 움직임 벡터 후보는 미디언 예측에 의해 결정된다. 그러나 세 개의 인접한 블록 중 한 개만 이용 가능하면 해당 매크로블록의 움직임 벡터로 결정된다^[4].

예측 움직임 벡터의 후보군 중에서 최적의 예측 움직임 벡터는 비트율-왜곡 최적화 기법을 통해 결정된다. 즉, 식 (3)의 최소값으로 최적의 예측 움직임 벡터가 결정된다.

$$J = D + \lambda(R_m + R_r + R_{mv} + R_{mm} + R_o) \quad (3)$$

여기서 J 는 비트율-왜곡 비용함수로 왜곡도와 함께 발생 비트를 고려한다. D 는 원 영상과 움직임 예측에 의해 움직임이 보강된 영상의 차이를 나타내고 λ 는 발생 비트수의 가중치 값인 Lagrangian 계수이다. R_m 은 매크로블록의 모드를 부호하는데 필요한 비트수이고, R_r 은 잔여 영상을 부호하는데 필요한 비트수이다. R_{mv} 은 움직임 벡터의 차분 값을 나타내는데 필요한 비트수이고 R_{mm} 은 예측 움직임 벡터를 나타내는 인덱스를 부호하는데 필요한 비트수이다. R_o 는 헤더 정보를 부호하는데 필요한 비트수이다.

III. 제안하는 예측 움직임 벡터 부호화

1. 예측 움직임 벡터의 분포

인터 모드는 시간적·공간적 특성을 고려한 후보 예측 움직임 벡터와 (0, 0) 벡터를 포함하여 총 6 개의 후보 벡터들이 경쟁한다. SKIP 모드의 경우에는 6 개의 후보 벡터와

확대된 공간적 예측 움직임 벡터가 경쟁한다. 선택된 예측 움직임 벡터의 분포도를 알아보기 위해 세 개의 CIF급 해상도를 갖는 영상 (Akiyo, Foreman, Mobile) 50 프레임을 이용하여 실험했다. 양자화 계수는 27, 37을 사용하였고 탐색 영역은 32로 문맥 기반 적응적 가변 길이 부호화를 이용하여 실험했다. 그림 3은 SKIP 모드에서 선택된 예측 움직임 벡터의 분포를 나타낸다. 그림 3에서 보듯이, SKIP 모드에서는 미디언 예측기를 통해 결정된 움직임 벡터가 최적의 예측 움직임 벡터로 선택되는 빈도가 다른 후보군보다 월등히 높은 것을 확인할 수 있다.

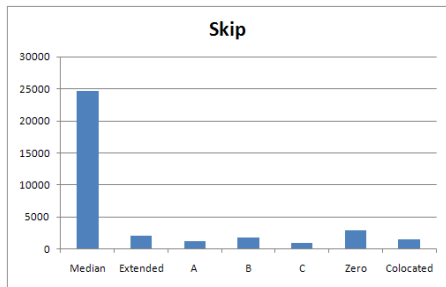


그림 3. SKIP 모드에서 최적 예측 움직임 벡터 분포도
Fig. 3. Distribution of the best motion vector predictor for the SKIP mode

그림 4는 인터 모드에서 선택된 예측 움직임 벡터의 분포도를 나타낸다. 인터 모드는 SKIP 모드와 달리 미디언 예측기를 통해 얻은 예측 움직임 벡터와 현재 매크로블록을 기준으로 참조 화면에서 동일한 위치에 있는 매크로블록의 움직임 벡터가 최적의 예측 움직임 벡터로 선택되는 빈도가 높은 것을 확인할 수 있다.

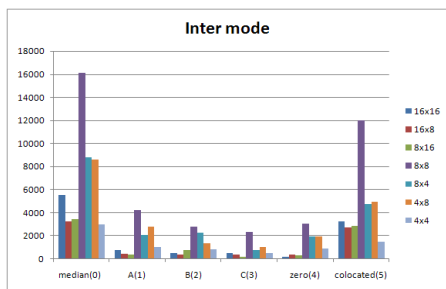


그림 4. 인터 모드에서 최적 예측 움직임 벡터 분포도
Fig. 4. Distribution of the best motion vector predictor for the inter mode

그림 3과 그림 4를 통해 예측 움직임 벡터의 분포가 균일하지 않다는 사실을 확인할 수 있다. 그러므로 고정 길이의 코드워드를 할당하는 방법보다 빈도수가 높은 예측 움직임 벡터의 인덱스에는 짧은 코드워드를 할당하고 빈도수가 낮은 예측 움직임 벡터의 인덱스에는 긴 코드워드를 할당하는 방법이 더 효율적이다.

2. Phased-in 코드를 이용한 예측 움직임 벡터 부호화

기존의 KTA에서는 예측 움직임 벡터의 인덱스를 부호화하기 위해 고정 길이의 코드워드를 할당한다. 각각의 인덱스 정보를 표현하기 위해 식 (4)를 통해 인덱스에 대한 코드의 길이가 결정된다. 사용자가 설정한 예측 움직임 벡터의 개수와 종류에 따라 고정 길이 코드워드를 할당하기 위해 식 (4)를 이용한다.

$$length = (maxmode / 2) + (maxmode \bmod 2) \quad (4)$$

인터 모드는 총 6 개의 예측 움직임 벡터 후보가 경쟁하므로 3 비트를 이용하여 예측 움직임 벡터의 인덱스 정보를 표현한다. SKIP 모드는 총 7 개의 예측 움직임 벡터 후보가 경쟁하므로 인터 모드와는 다르게 4 비트로 인덱스 정보를 표현한다. 식 (4)로 인해 SKIP 모드의 예측 움직임 벡터가 7 개임에도 불구하고 4 비트로 인덱스 정보를 표현한다.

이 논문에서는 인터 모드와 SKIP 모드에서 예측 움직임 벡터가 선택되는 빈도를 기반으로 Phased-in 코드를 이용하여 효율적인 코드워드 표를 설계한다⁵⁾.

Phased-in 코드는 서로 다른 2 개의 길이를 가진 코드워드로 구성된 것이다. 일반적으로 심볼들이 동일한 확률을 가지면 고정 길이의 코드를 할당하여 압축의 효과를 기대하기 어렵다. 그러나 Phased-in 코드는 심볼들이 동일한 확률을 가지더라도 서로 다른 길이를 가진 코드워드를 할당하여 압축의 효과를 얻을 수 있다. 또한 Phased-in 코드는 접두어 코드로 복호기에서 모호함이 없이 복호화 과정이 이루어진다.

다음의 과정은 인터 모드의 인덱스 코드 표를 설계하는 방법을 설명한다.

- 1) 인터 모드의 경우, 총 6 개의 예측 움직임 벡터를 사용하므로 부호해야 할 심볼은 6 개이고, n은 6이다.
- 2) m을 계산한다. m+1은 n 개의 심볼을 표현하기 위한 최소 비트수이다. 즉, $2^m \leq n < 2^{m+1}$ 을 만족해야 한다. 총 6 개의 심볼을 표현하기 위한 최소의 비트수는 3이므로 m은 2이다.
- 3) p와 P를 계산한다. p는 $n - 2^m$ 이고 P는 $2^m - p$ 이다. 2p와 P의 합은 전체 심볼의 개수와 같다. 인터 모드의 경우 p는 $2(=6 - 2^2)$ 이고, P는 $2(=2^2 - 2)$ 이다.
- 4) P보다 크기가 작은 심볼은 m 개의 비트를 가진 고정 길이 코드워드로 표현된다. 즉, 2 보다 작은 심볼은 2 비트로 표현된다.
- 5) 과정 4) 이외의 심볼은 m+1 개의 비트를 가진 고정 길이 코드워드로 표현된다. 고정 길이 코드워드의 값은 $2^{m+1} - 2p$ 에서 $2^{m+1} - 1$ 를 가진다. 즉, 인터 모드의 경우, 2부터 5인 인덱스 값이 4부터 7로 사상되고, 4부터 7의 값을 3 비트를 이용하여 나타낸다.

위 과정을 SKIP 모드에 관한 예측 움직임 벡터의 인덱스 코드 결정에도 적용하여, 효율적인 코드워드를 할당할 수 있다. SKIP 모드는 총 부호해야 할 심볼이 7 개이므로 m은 2이고, p는 3이고 P는 1이다. 1보다 작은 심볼은 2 비트로 표현되고 그 외의 심볼은 2부터 7로 사상되어 3 비트로 표현된다.

표 1. 인터 모드 예측 움직임 벡터 인덱스 코드워드
Table 1. The codeword of the motion vector predictor for the inter mode

예측 움직임 벡터	인덱스	기존의 코드워드	제안하는 코드워드
Median	0	000	00
A	1	001	100
B	2	010	101
C	3	011	110
Zero(0, 0)	4	100	111
Collocated	5	101	01

표 1은 인터 모드에서 예측 움직임 벡터의 인덱스를 부호하기 위해 Phased-in 코드를 이용하여 설계된 새로운 코드

워드를 나타낸다. 여기서 인덱스 0과 인덱스 5는 자주 발생하는 예측 움직임 벡터로 0과 1로 사상되어 2 비트의 고정 길이 코드워드가 할당되고, 인덱스 1부터 인덱스 4는 덜 발생하는 예측 움직임 벡터로 4부터 7로 사상되어 3 비트의 고정 길이 코드워드가 할당된다.

표 2. SKIP 모드 예측 움직임 벡터 인덱스 코드워드
Table 2. The codeword of the motion vector predictor for the SKIP mode

예측 움직임 벡터	인덱스	기존의 코드워드	제안하는 코드워드
Median	0	0000	00
Extended Spatial	1	0001	010
A	2	0010	011
B	3	0011	100
C	4	0100	101
Zero(0, 0)	5	0101	110
Collocated	6	0110	111

표 2는 SKIP 모드를 위한 예측 움직임 벡터의 인덱스 코드워드를 보여준다. SKIP 모드는 인덱스 0인 예측 움직임 벡터가 많이 발생하므로 0으로 사상하여 2 비트의 고정 길이 코드워드를 할당했고, 나머지 인덱스는 2부터 7로 사상되어 3 비트의 고정 길이 코드워드를 할당했다.

기존의 방법은 인터 모드의 예측 움직임 벡터 인덱스를 표현하기 위해 3 비트를 사용했지만, 제안하는 방법은 자주 발생하는 예측 움직임 벡터에는 2 비트를 할당하고 나머지 예측 움직임 벡터에는 3 비트를 할당하므로 평균 비트수를 줄일 수 있다. SKIP 모드도 4 비트를 사용했지만, 자주 발생하는 미디언 예측에 대해서는 2 비트를 할당하고, 나머지 비트에 대해서는 3 비트를 할당하여 평균 비트수를 줄여 압축 성능을 높일 수 있다.

IV. 실험 결과

이 논문에서는 KTA 2.6 소프트웨어에 제안한 알고리즘을 구현했다^[6]. 제안한 알고리즘의 성능을 보이기 위해 4개의 CIF급 (352×288) 해상도를 갖는 영상 (Akiyo, Fore-

man, Mobile, Paris), 3 개의 HD급 (1280×720) 해상도를 갖는 영상(Raven, Crew, Bigship) 100 프레임을 사용했다. 실험에 사용한 자세한 부호화 변수는 표 3에 나타낸다.

표 3. 실험 환경
Table 3. Encoding parameters

부호화 변수	값
<i>ProfileIDC</i>	66 (baseline profile)
<i>QP</i>	22, 27, 32, 37
<i>SearchRange</i>	32
<i>SymbolMode</i>	0 (CAVLC)
<i>Frame Structure</i>	IPPP...P
<i>MVCompetition</i>	2
<i>Predictors_skip</i>	11111110
<i>Predictors_MVp</i>	10111110

표 3에서 *MVCompetition*을 2로 설정하면 사용자가 예측 움직임 벡터의 개수와 종류를 조절할 수 있다. *Predictor_skip*은 SKIP 모드에서 각각의 예측 움직임 벡터의 사용 여부를 결정하는 것으로 본 실험에서는 총 7 개의 예측 움직임 벡터(Median, Extended Spatial, A, B, C, Zero, Collocated)를 이용했다. 인터 모드에서는 총 6 개의 예측 움직임 벡터(Median, A, B, C, Zero, Collocated)를 사용하도록 설정했다.

제안한 방법과 KTA 소프트웨어 방법의 성능 비교를 위해 BDBR(Bjonteggard Delta Bitrate)와 BDPSNR (Bjonteggard Delta PSNR)을 이용했다^[7]. 표 4에서 보는 바와 같이, 제안한 알고리즘은 기존의 KTA에 비해 BDBR로 평균 약 7.24%, BDPSNR로 평균 약 0.36dB의 부호화 이득을 제공했다. 일반적으로 다양한 움직임 벡터 예측 방법이 H.264/AVC보다 우수한 성능을 내기 위해서는 예측 움직임 벡터를 2~4개 사용해야 한다. 예측 움직임 벡터의 종류가 증가할수록 움직임 벡터 차분값을 부호하는데 소요되는 비트의 양은 감소하지만, 반대로 복호기로 전송해야 할 인덱스 정보가 증가하여 압축 효율이 감소되기 때문이다.

이 논문은 H.264/AVC와 비교하기 위해 예측 움직임 벡터를 3개 사용하여 실험을 수행했다. 예측 움직임 벡터는

발생 빈도가 높은 순서로 인터 모드에서는 *mvmedian*, *mva*, *mvcollocated*를 선택했고, SKIP 모드에서는 *mvmedian*, (0, 0), *mvcollocated*를 선택했다. 예측 움직임 벡터의 인덱스를 부호하기 위해 사용한 코드워드는 표 5와 같다. 표 6에서 보듯이, 제안하는 방법은 H264/AVC에 비해 BDBR로 평균 약 1.30%, BDPSNR로 평균 약 0.06dB의 부호화 이득을 제공했다.

표 4. 제안한 알고리즘과 KTA 성능 비교
Table 4. Comparison of BDPSNR and BDBR

실험 영상	BDBR(%)	BDPSNR(dB)
Foreman	-5.80	0.28
Akiyo	-12.15	0.82
Mobile	-2.02	0.11
Paris	-3.39	0.20
Crew	-5.58	0.20
Raven	-13.19	0.66
Bigship	-8.53	0.33
평균	-7.24	0.36

표 5. 인터 모드와 SKIP 모드 예측 움직임 벡터 인덱스 코드워드
Table 5. The codeword of the motion vector predictor for the inter mode and the SKIP mode

인터모드		SKIP 모드	
예측 움직임 벡터	제안하는 코드워드	예측 움직임 벡터	제안하는 코드워드
Median	0	Median	0
A	10	Zero(0, 0)	10
Collocated	11	Collocated	11

표 6. 제안한 알고리즘과 H.264/AVC 성능 비교
Table 6. Comparison of H.264/AVC and proposed algorithm

실험 영상	BDBR(%)	BDPSNR(dB)
Foreman	-1.64	0.07
Akiyo	-1.84	0.08
Mobile	-1.63	0.09
Paris	-1.21	0.07
Crew	-0.50	0.01
Bigship	-0.98	0.03
평균	-1.30	0.06

그림 5는 제안하는 방법과 기존의 KTA 방법에 대한 비트율-왜곡 곡선으로 양자화 계수가 22, 27, 32, 37인 경우에 대한 결과를 보여준다. 특히, 제안한 알고리즘은 양자화 계수가 큰 저비트율 환경에서 더 좋은 성능을 보여주는 것을 확인할 수 있다. 그림 5에서 보듯이, 이 논문에서 제안하는 알고리즘은 Akiyo나 Raven과 같은 특성이 단순한 영상에서 더 좋은 성능을 보인다. 단순한 특성을 지닌 영상에서는 SKIP 모드나 16×16 인터 모드가 많이 선택되지만, 복잡한 특성을 지닌 영상에서는 P8×8 모드가 많이 선택된다.

제안하는 방법은 SKIP 모드나 16×16 인터 모드를 통해 절감되는 비트보다 P8×8 모드를 통해 절감되는 비트의 양이 더 많기 때문에 P8×8 모드가 최적의 매크로블록 모드로 많이 선택된다. 따라서 단순한 영상과 복잡한 영상에서 화질은 증가하지만 P8×8 모드로 선택되는 매크로블록의 개수가 증가하므로 P8×8 모드로 결정되는 매크로블록에 대해 보내야 할 정보양이 많다. 그러므로 Mobile과 같은 복잡한 영상에서는 예측 움직임 벡터의 인덱스를 표현하는 비트를 줄여도 P8×8 모드로 결정된 매크로블록에 대해 보내야 할 정보의 양이 많기 때문에 복잡한 특성을 가진 영상은 단순한 영상에 비해 좋은 성능을 보이지 못한다.

V. 결론

이 논문에서는 다양한 예측 움직임 벡터를 이용한 움직임 벡터 정보 부호화 방법을 향상시키기 위해 효율적인 예측 움직임 벡터 인덱스 부호화 방법을 제안했다. 기존의 다양한 움직임 벡터를 이용한 움직임 벡터 부호화 방법은 다양한 예측 움직임 벡터 후보를 사용하여 실제 부호할 움직임 벡터 차분값의 크기를 줄여 움직임 벡터 정보를 부호하는 비트를 절감할 수 있다. 하지만, 복호를 위해 선택된 예측 움직임 벡터의 인덱스 정보를 추가로 부호하기 때문에, 압축 효율이 감소되는 문제점이 있다. 이 논문에서는 추가로 부호되는 예측 움직임 벡터의 인덱스 정보를 효율적으로 부호화하기 위해, 예측 움직임 벡터의 선택 빈도에 따라서 서로 다른 길이의 코드워드를 할당했다. 제안한 방법은 인터 모드에서는 미디언 예측 움직임 벡터와 참조 화면에서 동일한 위치에 있는 매크로블록의 움직임 벡터가 예측 움

직임 벡터로 많이 선택되므로 짧은 코드워드를 할당했다. 또한, SKIP 모드에서는 미디언 예측기를 통해 결정된 예측 움직임 벡터가 많이 발생하므로 해당 인덱스에 짧은 코드워드를 할당했다. 제안한 방법을 이용하면 기존의 KTA의 다양한 움직임 벡터 예측 방법에 비해, BDBR로 평균 약 7.24%, BDPSNR로 평균 약 0.36dB의 부호화 성능이 향상됨을 확인했다.

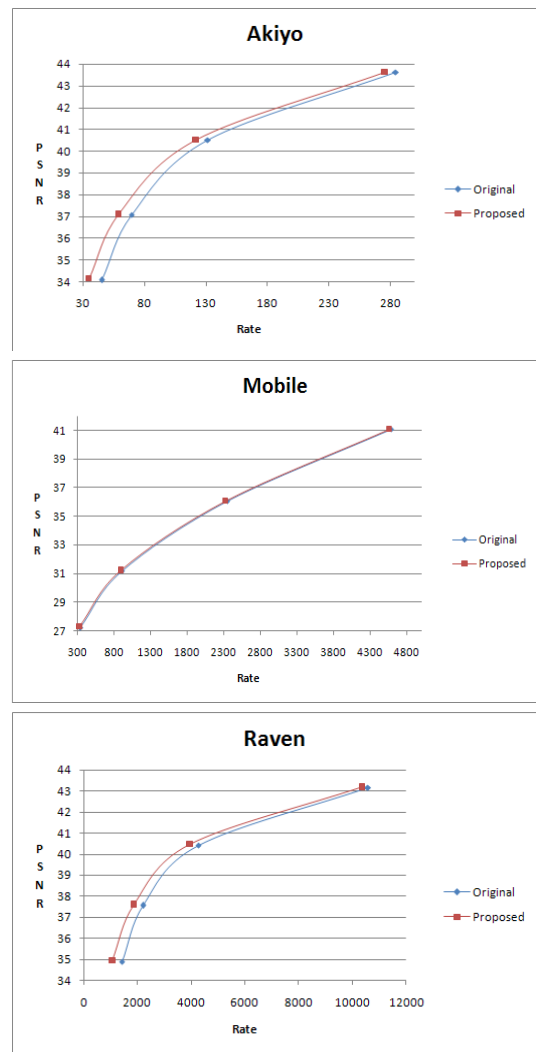


그림 5. 비트율-왜곡 곡선
Fig. 5. Rate distortion curves

참 고 문 헌

[1] Competition-Based Scheme for Vector Selection and Coding, document VCEG-AC06.doc, ITU-T SG16/Q6, July 2006.

[2] 호요성, 김승환, H.264/AVC 알고리즘 이해와 프로그램 분석, 두양사, 2009.

[3] D. Marpe, T. Wiegand, and G. Sullivan, "The H.264/MPEG4 Advanced Video Coding Standard and Its Application," IEEE Communications Magazine, vol. 44, no. 8, pp. 134-144, Aug. 2006.

[4] G. Laroche, J. Jung, and B. Pesquet-Popescu, "Competition Based Prediction for SKIP Mode Motion Vector Using Macroblock Classification for the H.264 JM KTA Software," Lecture Notes in Computer Science, vol. 4678, pp. 789-799, Aug. 2007.

[5] D. Salomon, Variable-length Codes for Data Compression, Springer, 2007.

[6] <http://iphome/hhi.de/shehring/tml/download/KTA/jm11.0kta2.6.r1.zip>, KTA software version 2.6. r1.

[7] Improvement of the BD-PSNR Model, document VCEG-A11 1.doc, ITU-T SG16/Q6, July 2008.

저 자 소 개



문 지 희

- 2005년 ~ 2008년 : 세종대학교 전자공학과 학사
- 2009년 ~ 현재 : 광주과학기술원 정보기전공학부 정보통신공학과 석사과정
- 주관심분야 : 디지털 영상처리, H.264/AVC, High Efficiency Video Coding



최 정 아

- 2002년 ~ 2007년 : 한국항공대학교 전자 및 항공전자공학과 학사
- 2007년 ~ 2008년 : 광주과학기술원 정보기전공학부 정보통신공학과 석사
- 2008년 ~ 현재 : 광주과학기술원 정보기전공학부 정보통신공학과 박사과정
- 주관심분야 : 디지털 영상처리, H.264/AVC, High Efficiency Video Coding



호 요 성

- 1977년 ~ 1981년 : 서울대학교 전자공학과 학사
- 1981년 ~ 1983년 : 서울대학교 전자공학과 석사
- 1983년 ~ 1995년 : 한국전자통신연구소 선임연구원
- 1985년 ~ 1989년 : University of California, 전기전산공학과 박사
- 1990년 ~ 1993년 : 미국 필립스 연구소 선임연구원
- 1995년 ~ 현재 : 광주과학기술원 정보통신공학부 교수
- 2003년 ~ 현재 : 광주과학기술원 실감방송연구센터 센터장
- 주관심분야 : 디지털 영상신호 처리 및 압축, 디지털 TV, MPEG 표준, 3차원 TV, 실감방송