일반논문 (Regular Paper)
방송공학회논문지 제17권 제6호, 2012년 11월 (JBE Vol. 17, No. 6, November 2012)
http://dx.doi.org/10.5909/JBE.2012.17.6.1029
ISSN 1226-7953(Print)

HEVC에서의 적응적 움직임 벡터 해상도를 이용한 움직임 추정 및 부호화 기법

임 성 원[a], 이 주 옥[a], 문 주 희[a]‡

Motion Estimation and Coding Technique using Adaptive Motion Vector Resolution in HEVC

Sung-Won Lim[a], Ju Ock Lee[a], and Joo-Hee Moon[a]‡

요 약

본 논문에서는, 인터 예측시 움직임 벡터의 해상도를 나타내는 1비트 플래그를 두어 적응적으로 1/4 해상도의 움직임 벡터와 1/8 해상도의 움직임 벡터를 선택하고 부호화하는 방법을 제안한다. 현재 HEVC에서는 1/4 해상도의 움직임 벡터만을 이용하여 부호화하는데, 영상 신호의 변화가 복잡한 영역에서 1/4 해상도의 움직임 벡터만으로는 충분한 효율을 얻어내지 못한다. 따라서 본 논문에서는 PU마다 해상도 플래그를 1비트 추가하여 적응적으로 움직임 벡터의 해상도를 결정할 수 있도록 한다. 제안한 방법의 실험 결과로서, 인코더의 복잡도는 30%~33% 증가하고 디코더의 복잡도는 1%~5% 증가하였지만, 휘도신호의 압축효율은 최대 5.3% 좋아졌으며, 색차신호의 압축효율은 최대 7.9% 좋아졌다.

Abstract

In this papar, we propose a new motion estimation and coding technique using adaptive motion vector resolution. Currently, HEVC encodes a video using 1/4 motion vector resolution. If there are high texture regions in a picture, HEVC can't get a performance enough. So, we insert additional 1-bit flag meaning whether motion vector resolution is 1/4 or 1/8 in PU syntax. Therefore, decoder can recognize the transmitted motion vector resolution. Experimental results show that maximum coding efficiency gain of the proposed method is up to 5.3% in luminance and 7.9% in chrominance. Average computional time complexity is increased about 33% in encoder and up to 5% in decoder.

Keyword : HEVC, Inter prediction, Motion vector estimation, Adaptive motion vector resolution

a) 세종대학교 정보통신연구소 정보통신공학과(Sejong University, Information and Telecommunication Research Institute, Dept. of Information and Communications Engineering)
‡ Corresponding Author : 문주희 (Joo-Hee Moon)
E-mail: jhmoon@sejong.ac.kr
Tel: +82-2-3408-3829 Fax: +82-2-3408-4330
· Manuscript received July 20, 2012 Revised September 3, 2012 Accepted November 23, 2012

Ⅰ. 서 론

최근 동영상의 효율적인 압축을 위해 ISO/IEC MPEG (Moving Picture Experts Group)과 ITU-T VCEG (Video Coding Experts Group)에서 공동으로 JCT-VC (Joint

Collaborative Team on Video Coding)을 결성하였다. JCT-VC에서는 종전의 표준인 H.264/AVC[1]에 비해 약 2배의 부호화 효율을 목표로 HEVC(High Efficiency Video Coding)의 개발에 대한 표준화 연구를 진행하여, 2012년 2월 CD(Committee Draft: 위원회표준안)[2]이 제작되었다.

H.264/AVC에서는 하나의 slice를 16x16크기의 매크로블록(MB: Macro Block)으로 나누고 16x16~4x4 크기로 분할하여 부호화하는 방식을 가지는데 비해, HEVC에서는 하나의 slice를 64x64 크기의 LCU(Largest Coding Unit)으로 나눈 후 각각의 LCU를 64x64~8x8 단위의 쿼드트리(quad tree) 형태의 CU(Coding Unit)로 분할하여 부호화한다. 각 CU에서 인트라 예측시에는 2Nx2N과 NxN으로, 인터 예측시에는 2Nx2N~NxN 및 AMP[3](Asymmetric Motion Partition: 2NxnU, 2NxnD, nLx2N, nRx2N)를 이용하여 해당하는 PU(Prediction Unit)로 분할하여 예측하는 방식을 가진다.

각 CU가 인터 예측을 위한 PU로 분할되었을 때, 움직임 추정(ME: Motion Estimation)과정을 통하여 현재 픽쳐와 인접한 주위 픽쳐에서 현재 부호화하려는 PU와 가장 유사한 예측 블록을 찾게 되고, 현재 블록과 예측 블록간의 움직임 정보를 나타내는 움직임 파라미터(motion parameter)를 추정한다. 현재 CU는 최적으로 예측할 수 있도록 분할한 PU들의 조합이 결정된 후, 변환, 양자화, 엔트로피 부호화 과정을 거쳐 압축된다.

HEVC에서는 움직임 추정시 움직임 벡터를 정수 화소 위치로 찾은 후, 픽쳐의 해상도를 4배 확대하여 1/4 화소 위치까지 움직임 벡터를 구하게 된다. 하지만 영상의 특성에 따라 1/4 화소 위치로 움직임 벡터를 구하는 것이 충분하지 못할 수도 있으며, 반대로 1/8 화소 위치로 움직임 벡터를 구하는 것이 1/4 화소 위치로 움직임 벡터를 구하는 것보다 비효율적일 수도 있다[4]. 일반적으로 영상에 많은 텍스쳐(texture)가 존재할 때는 해상도를 높이는 것이 좋으며, 영상에 잡음이 많이 끼거나 텍스쳐가 적을 때는 해상도를 낮게 하는 것이 좋다고 알려져 있다[5]. 따라서 본 논문에서는, 움직임 벡터의 해상도를 나타내는 비트를 이용하여 어느 해상도가 최적인지를 적응적으로 판단하고 부호화하는 방법을 제안한다.

본 논문의 Ⅱ장에서는 기존의 움직임 추정 방식에 대해서 설명을 하고, Ⅲ장에서는 제안하고 있는 적응적 움직임 벡터 해상도에 대해서 설명을 하며, Ⅳ장에서 실험 결과를 보인 후, 마지막으로 Ⅴ장에서 결론을 기술한다.

Ⅱ. HEVC의 움직임 벡터 추정 방법

1. 정수 화소 위치 움직임 추정

HEVC에서는 인터 예측시 각 CU마다 다양한 모양의 PU를 이용하여 움직임 보상 예측을 수행한다. <그림 1>은 움직임 벡터를 찾는 과정을 그림으로 표현하기 위해 현재 PU가 4x4라고 가정을 한 예이다. 그림에서와 같이 현재 픽쳐와 인접한 주위 픽쳐에서 가장 유사한 블록인 예측 블록을 찾은 후 잔차신호를 구한다. 이때 잔차신호

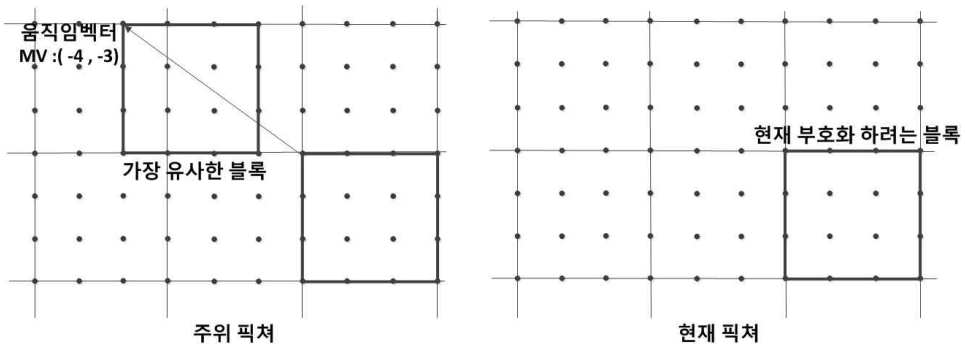

그림 1. 정수 화소 위치 움직임 추정의 예
Fig. 1. An example of motion estimation for integer position

는 SAD(Sum of Absolute Difference)를 이용하며, SAD와 움직임 파라미터의 비트를 고려하여 최적의 움직임 벡터를 찾게 된다.

2. 분수 화소 위치 움직임 추정

분수 화소 위치로 움직임 보상을 할 경우에는 정수 화소 위치에서 구한 움직임 벡터를 기준으로 분수 화소 위치로 조사를 하여 유사한 블록이 있는지 찾게 된다. HEVC는 1/4 화소 위치까지 움직임을 추정하는데, 1/2 화소 위치 움직임 벡터를 구한 후 1/4 화소 위치 움직임 벡터를 순차적으로 구하게 된다. 정수 화소 위치에서 움직임 벡터를 구한 후, 해상도를 2배로 확대하여 정수 화소 위치에서 구한 움직임 벡터를 기준으로 (-1,-1) ~ (1,1) 범위에서 가장 유사한 예측 블록을 찾게 된다. 다시 해상도를 2배로 확대하여 4배의 해상도에서 1/2 화소 위치 움직임 벡터를 기준으로 (-1,-1) ~ (1,1) 범위에서 가장 유사한 예측 블록을 찾음으로서, 최종적인 4배의 해상도에서 구한 움직임 벡터를 얻게 된다. <그림 2>에서 1/2 화소 위치 움직임 벡터 추정과 1/4 화소 위치 움직임 벡터의 추정에 대한 예시가 있다. <그림 1>에서의 움직임 벡터 (-4,-3)는 2배의 해상도에서는 (-8,-6)이 되고 (0,-1) 만큼 움직였으므로 2배의 해상도에서 최종적으로 (-8,-7)을 얻게 된다. 4배의 해상도에서 움직임 벡터는 (-16,-14)으로 변경이 되고, (-1,0)만큼 움직였으므로 최종적인 움직임 벡터 (-17,-14)가 나오게 된다.

3. 분수 화소 위치를 위한 보간 방법

HEVC에서는 1/2 위치 화소와 1/4 위치 화소를 위해 DCT-IF(Dicrete Cosine Transform-based Interpolation Filter)를 사용하고 있다. <그림 3>에서 A, B, C, D는 정수 위치 화소들이고, a~o는 분수 위치 화소들이다. a,b,c와 d,h,l은 1D 보간 필터를 이용하여 보간되며, 그외의 분수 위치 화소들은 2D 보간 필터를 이용하여 보간된다. 2D 보간 필터 방식은, 세로로 1D 보간 필터를 사용하여 나온 보간값들에 다시 가로로 1D 보간 필터를 사용하는 방식이다. 1D 보간 필터의 설명은 <그림 4>에 나와있다. HEVC에서는 1/2 위치 화소의 보간을 위해 왼쪽 4탭과 오른쪽 4탭을 이용하는 8탭 필터를 사용하며, 1/4와 3/4 위치 화소의 보간을 위해 7탭 필터를 사용한다. <그림 4>에서 x_0~x_7은 정수 화소 위치를 표현하며, x_{q0}, x_h, x_{q1}들은 각각 1/4, 1/2, 3/4

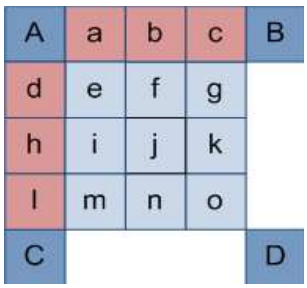

그림 3. 정수 위치 화소와 분수 위치 화소들
Fig. 3. Pixels at integer and fractional pixel position

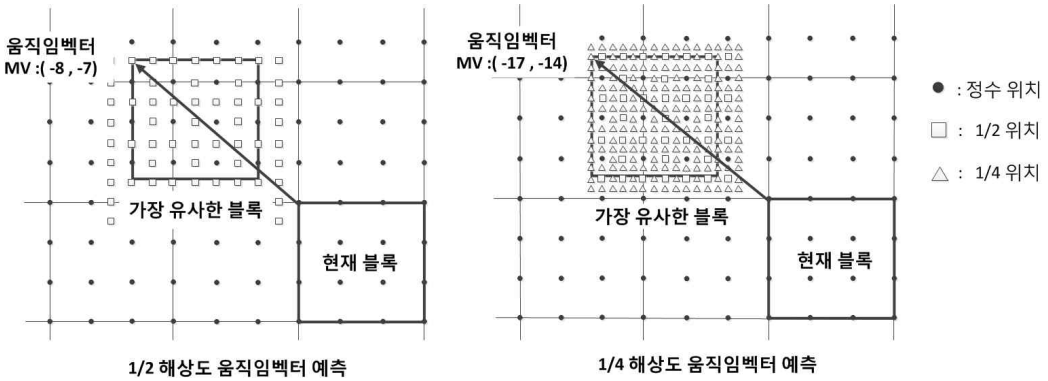

그림 2. 1/2 화소 위치 움직임 벡터 추정(왼쪽)과 1/4 화소 위치 움직임 벡터 추정(오른쪽)
Fig. 2. Motion estimation at 1/2 pixel position (left) and 1/4 pixel position (right)

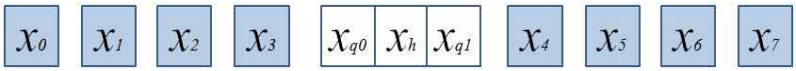

그림 4. 보간을 위한 정수 위치 화소와 분수 위치 화소들
Fig. 4. Pixels at integer and fractional pixel position for interpolation

화소 위치를 나타낸다. 분수 위치 화소를 보간하기 위해서 주위 정수 화소 위치의 화소들을 이용하는데, 예를 들어 1/2 화소 위치를 보간할 경우 식 (1)과 같이 표현될 수 있다.

$$x_h = C_0x_0 + C_1x_1 + C_2x_2 + C_3x_3 + C_4x_4 + C_5x_5 + C_6x_6 + C_7x_7 \tag{1}$$

여기서 C_0~C_7은 보간 필터 계수들이다. 해당하는 보간 필터 계수들은 다음에 설명되는 바와 같이 DCT(Discrete Cosine Transform)와 IDCT(Inverce DCT)식을 이용하여 유도한다.

유한개의 입력 데이터 $\{x(n), 0 \le n \le N-1\}$에 대한 1차원 DCT와 IDCT는 각각 식 (2)와 식 (3)과 같이 정의된다.

$$DCT \quad : \quad y(k) = \alpha(k)\sum_{n=0}^{N-1} x(n)\cos[\frac{\pi k(2n+1)}{2N}], \quad 0 \le k \le N-1 \tag{2}$$

$$IDCT \quad : \quad x(n) = \sum_{k=0}^{N-1} \alpha(k)y(k)\cos[\frac{\pi k(2n+1)}{2N}], \quad 0 \le n \le N-1 \tag{3}$$

여기서, $\alpha(0) = \frac{1}{N}$, $\alpha(k) = \sqrt{\frac{2}{N}}$ 이며, 보간 필터 계수를 유도하기 위해 식 (3)의 IDCT를 다시 한번 정리하게 되면 식(4)로 정리가 된다.

$$\begin{aligned} x(n) = {} & \alpha(0)y(0)\cos\frac{\pi\times0\times(2n+1)}{2N} + \\ & \alpha(1)y(1)\cos\frac{\pi\times1\times(2n+1)}{2N} + \cdots + \\ & \alpha(N-1)y(N-1)\cos\frac{\pi\times(N-1)\times(2n+1)}{2N} \end{aligned} \tag{4}$$

식 (4)에서 $y(0)$~$y(N-1)$에 대해 식 (2)를 대입하면 식 (5)로 정리가 된다.

식 (5)를 $x(0)$~$x(N-1)$으로 정리하면 식 (6)이 되며,

$$\begin{aligned} x(n) = {} & \alpha^2(0)\,[\,x(0)\cos\frac{\pi\times0\times(2\times0+1)}{2N} + \cdots + x(N-1)\cos\frac{\pi\times0\times(2\times(N-1)+1)}{2N}\,]\,\cos\frac{\pi\times0\times(2n+1)}{2N} + \\ & \alpha^2(1)\,[\,x(0)\cos\frac{\pi\times1\times(2\times0+1)}{2N} + \cdots + x(N-1)\cos\frac{\pi\times1\times(2\times(N-1)+1)}{2N}\,]\cos\frac{\pi\times1\times(2n+1)}{2N} + \cdots + \\ & \alpha^2(N-1)[x(0)\cos\frac{\pi\times(N-1)\times(2\times0+1)}{2N} + \cdots + \\ & x(N-1)\cos\frac{\pi\times(N-1)\times(2\times(N-1)+1)}{2N}\,]\,\cos\frac{\pi\times(N-1)\times(2n+1)}{2N} \end{aligned} \tag{5}$$

$$\begin{aligned} x(n) = {} & x(0)[\alpha^2(0)\cos\frac{\pi\times0\times(2\times0+1)}{2N}\cos\frac{\pi\times0\times(2n+1)}{2N} + \cdots + \\ & \alpha^2(N-1)\cos\frac{\pi\times(N-1)\times(2\times0+1)}{2N}\cos\frac{\pi\times(N-1)\times(2n+1)}{2N}\,] + \\ & x(1)[\alpha^2(0)\cos\frac{\pi\times0\times(2\times1+1)}{2N}\cos\frac{\pi\times0\times(2n+1)}{2N} + \cdots + \\ & \alpha^2(N-1)\cos\frac{\pi\times(N-1)\times(2\times1+1)}{2N}\cos\frac{\pi\times(N-1)\times(2n+1)}{2N}\,] + \cdots + \\ & x(N-1)[\alpha^2(0)\cos\frac{\pi\times0\times(2\times(N-1)+1)}{2N}\cos\frac{\pi\times0\times(2n+1)}{2N} + \cdots + \\ & \alpha^2(N-1)\cos\frac{\pi\times(N-1)\times(2\times(N-1)+1)}{2N}\cos\frac{\pi\times(N-1)\times(2n+1)}{2N}\,] \\ = {} & x(0)\,C_0(n) + x(1)\,C_1(n) + \cdots + x(N-1)\,C_{N-1}(n) \end{aligned} \tag{6}$$

실수 값을 대입하기 위해 정수 n을 실수 m으로 교체하면 최종적인 식 (7)이 된다.

$$x(m) = x(0)C_0(m) + x(1)C_1(m) + \cdots + x(N-1)C_{N-1}(m) \quad (7)$$

여기서 1/2 화소 위치 보간 필터 계수를 유도하기 위해서, 식 (1)의 경우에서와 같이 왼쪽 4탭과 오른쪽 4탭을 이용한다. 그러한 이유로, 식 (7)에서 보간 계수 $C(m)$를 유도할 경우, m 위치에 1/2 이 아닌 7/2를 대입하여 유도하여야 한다. 또한, 1/4과 3/4 위치 화소의 보간을 위해 식 (7)을 바탕으로 위상 변위(phase shift)를 이용하여 유도한 7탭 필터를 사용하고 있다[6]. 그 결과, 휘도와 색차에 따라 유도된 보간 필터 계수들이 각각 <표 1>과 <표 2>에 나와있다.

표 1. DCT-IF 계수(휘도)
Table 1. DCT-IF coefficients(luminance)

1/4	-1, 4, -10, 58, 17, -5, 1, 0
1/2	-1, 4, -11, 40, 40, -11, 4, -1
3/4	0, 1, -5, 17, 58, -10, 4, -1

표 2. DCT-IF 계수(색차)
Table 2. DCT-IF coefficients(chrominance)

1/8	-2, 58, 10, -2
1/4	-4, 54, 16, -2
3/8	-6, 46, 28, -4
1/2	-4, 36, 36, -4
5/8	-4, 28, 46, -6
3/4	-2, 16, 54, -4
7/8	-2, 10, 58, -2

III. 제안된 적응적 움직임 벡터 해상도를 이용한 부호화 기법

1. 1/8 화소 위치를 위한 보간 필터 계수

해상도를 8배로 확대하고 더욱 정밀한 1/8 화소 위치 움직임 벡터를 찾기 위해, 1/8 화소 단위까지 보간을 하여야 한다. <표 3>은 1/8 화소 위치를 위한 보간 필터 계수이다.

표 3. 1/8 화소 위치 보간을 위한 DCT-IF 계수
Table 3. DCT-IF coefficients at 1/8 pixel position for interpolation

1/8	-1, 3, -6, 62, 9, -4, 2, -1
1/4	-1, 4, -10, 58, 17, -5, 1, 0
3/8	-2, 5, -12, 50, 30, -10, 4, -1
1/2	-1, 4, -11, 40, 40, -11, 4, -1
5/8	-1, 4, -10, 30, 50, -12, 5, -2
3/4	0, 1, -5, 17, 58, -10, 4, -1
7/8	-1, 2, -4, 9, 62, -6, 3, -1

1/4 화소 위치를 위한 보간 계수들은 기존의 HEVC 방식을 그대로 사용하였고 1/8 화소 위치에 대해서는 1/4 화소 위치를 위한 계수들과 마찬가지로 1/8 화소 위치를 위한 DCT-IF 계수들을 유도하였다[7].

2. 1/8 화소 위치 움직임 추정

<그림 2>의 방법과 동일하게, 1/4 화소 위치로 구한 움직임 벡터를 기준으로 해상도를 다시 2배 확대 후 (-1,-1) ~ (1,1) 범위에서 가장 유사한 예측 블록을 찾게 된다. HEVC는 기본 해상도를 4배로 설정하는데 반해 본 논문에서는 8배로 설정한다.

3. 적응적 움직임 벡터 해상도

1/4 화소 위치 움직임 벡터와 1/8 화소 위치 움직임 벡터 중에서, 움직임 벡터를 부호화하는데 필요한 비트량과 잔차블록에서 계산된 SAD(Sum of Absolute Difference)를 이용하여 cost를 계산 후 알맞은 해상도를 결정하게 되며, cost를 구하는 방식은 기존의 방식에 해상도 플래그(resolution flag)를 추가한 방식이다. 어떤 해상도로 부호화 되었는지를 가리키는 해상도 플래그를 부호화하며 이때 식(8)을 이용하여 부호화하고, 식(9)를 이용하여 복호화한다. 여기서 MV(Motion Vector)는 부호화하려는 움직임 벡터, MVP(Motion Vector Prediction)는 예측 움직임 벡터를 나타내고, MVD(Motion Vector Difference)는 움직임 벡

터와 예측 움직임 벡터의 차이를 나타내게 되며, $MVRes$는 해상도 플래그를 나타낸다. MV와 MVP는 모두 1/8 화소 위치 움직임 벡터를 나타내며 1/4 화소 위치 움직임 벡터를 사용할 경우, MVD(Motion Vector Difference)는 1/4 화소 위치 기준으로 구하게 된다.

$$\begin{array}{ll} MV/2 - MVP/2 = MVD\,, & \text{if } MVRes \equiv 1/4 \\ MV - MVP = MVD\,, & otherwise \end{array} \quad (8)$$

$$\begin{array}{ll} MV = MVD \times 2 + (MVP/2) \times 2\,, & \text{if } MVRes \equiv 1/4 \\ MV = MVP + MVD\,, & otherwise \end{array} \quad (9)$$

위 식에서 움직임 벡터 성분 (x,y) 중 어느 하나라도 1/8 해상도를 가리키면 그 움직임 벡터의 해상도는 1/8로 설정이 되며, 두 성분 모두 1/4 화소 위치를 가리킬 때만 해상도를 1/4로 설정한다. 움직임 벡터가 1/4 위치 화소를 가리킬 경우, 식 (8)과 같이, 움직임 벡터와 예측 움직임 벡터를 1/4 화소 위치로 맞춰준 후 움직임 벡터 차이를 구한다. 이럴 경우, 디코더로 움직임 벡터 차이를 전송할 때, 1/8단위로 보내는 것보다 움직임 벡터 비트량을 대략 1/2로 보낼 수 있게 된다. 이때, 디코더에서는 움직임 벡터 차이를 다시 2배 하여 1/8 화소 위치로 설정해 준 후 예측 움직임 벡터를 더하여 움직임 벡터를 복원하게 된다. <표 4>는 HEVC에서 사용하고 있는 움직임 파라미터(motion parameter)를 전송하는 방식과 제안된 알고리즘의 syntax 비교이다. 여기서 mvr_flag는 식 (8)과 식 (9)에서 표현된 $MVRes$를 의미한다.

표 4 HEVC와 제안된 알고리즘의 PU syntax 비교
Table 4 A comparison of PU syntax with HEVC and proposed algorithm

현재 HEVC syntax	제안된 알고리즘의 syntax
.... ref_idx mvd mvp_idx ...	.... ref_idx mvr_flag mvd mvp_idx ...

4. 색차신호를 위한 1/16 화소 위치 보간 및 움직임 보상

HEVC는 기본적으로 4:2:0 표본형식을 사용하므로 색차 신호를 위한 움직임 보상시 휘도에서 사용된 움직임 벡터를 다운샘플링 후 사용한다. 휘도 신호에 적용된 1/8 화소 위치 움직임 벡터를 색차 신호에 적용을 하게 되면 다운샘플링 후 1/16 화소 위치 움직임 벡터가 된다. 휘도 신호 보간을 위해 식(2)~식(7)을 이용하여 보간 필터 계수를 유도해 낸 것과 마찬가지로 색차 신호 역시 보간 필터 계수를 동일하게 유도한다. <표 5>는 색차 신호를 위한 보간 필터 계수이다. 여기서 1/8 화소 위치는 기존의 보간 필터 계수를 그대로 사용하였다.

표 5. DCT-IF 계수 (색차)
Table 5. DCT-IF coefficients for chrominance

1/16	-2, 63, 4, -1
1/8	-2, 58, 10, -2
3/16	-5, 59, 13, -3
1/4	-4, 54, 16, -2
5/16	-7, 53, 23, -5
3/8	-6, 46, 28, -4
7/16	-7, 43, 34, -6
1/2	-4, 36, 36, -4
9/16	-6, 34, 43, -7
5/8	-4, 28, 46, -6
11/16	-5, 23, 53, -7
3/4	-2, 16, 54, -4
13/16	-3, 13, 59, -5
7/8	-2, 10, 58, -2
15/16	-1, 4, 63, -2

5. 해상도 플래그 유추를 통한 조건부 해상도 플래그 전송 기법

해상도 플래그를 전송하지 않고도 디코더에서 해상도 플래그를 유추할 수 있는 경우가 존재한다. 인코더에서 움직임 벡터 차이를 디코더로 전송할 때, 디코더 측면에서 보면, 전송되어온 움직임 벡터 차이가 짝수이고 주변 예측 움직임 벡터가 짝수이면 식(9)에 의하여 복원될 움직임 벡터는 해상도 플래그가 없이도 짝수가 되는 경우밖에 존재하지 않으므로 1/4 화소 위치 움직임 벡터를 사용했다는 것을 유추할 수 있다. 이 방식을 적용하기 위하여, <표 4>의 syntax에서 움직임 벡터 해상도를 나타내는 플래그를 제일 마지막에 보내는 것으로 변경할 수 있다. 이 기법에 대한 의사코드(pseudo code)가 <표 6>에 나타나 있다.

표 6. 해상도 플래그 유추를 위한 의사코드
Table 6. Pseudo-code for inferring a resolution flag

Encoder side	Decoder side

If (MVP == EVEN && MVD == EVEN)
Continue; // 1/4 resolution
Else
Encode (mvr_flag);
	If (MVP == EVEN && MVD == EVEN)
	mvr_flag = 1/4; // 1/4 resolution
	Else
	Decode (mvr_flag);

부호화 측면에서 보는 경우, MVP와 MVD가 전부 짝수이면 해상도 플래그를 부호화하지 않으며, 전부 짝수가 아닌 경우에만 해상도 플래그를 부호화한다. 복호화 측면에서 보는 경우 수신된 MVP와 MVD가 전부 짝수이면 해상도 플래그를 따로 복호화하지 않고도 현재 해상도가 1/4 화소 위치 움직임 벡터를 사용하는지 여부를 알 수 있으며, 전부 짝수가 아닐 경우에만 해상도 플래그를 복호화하여 현재 해상도를 결정한다.

Ⅳ. 실험 결과

1. 실험 조건

제안한 방법의 효과를 검증하기 위해, HM 6.0[8]에 제안한 방법을 구현하고, 압축 효율 및 복잡도를 비교하였다[9]. 또한 JCT-VC에서 사용하는 common condition[10]의 Random access main, Random access high efficiency, Low delay B main, Low delay B high efficiency의 총 4가지 설정을 사용하였으며, 평가 영상으로는 common condition에 나와있는 B~D class test sequence를 사용하였다. 그 외의 실험조건은 전부 common condition을 준수하였다. 또한, HM6.0을 기준으로 제안한 기술의 상대적 성능 변화를 측정하였으며 BD-rate 값이 음수일 경우 압축효율이 상승하였음을 의미하고 복잡도가 100%보다 작을경우 그만큼 복잡도가 줄어들었음을 의미한다.

2. 성능 평가

<표 7>은 1비트 해상도 플래그를 항상 이용한 방법의 효과를 평가한 결과이다. HM6.0과 비교하여 전체적으로 압축효율이 증가하는 것을 볼 수 있다. 하지만 HM6.0이 기존의 1/4 화소 위치 단위로 움직임 예측을 하는데 반해, 본 기술은 1/8 화소 위치 단위까지 움직임 예측을 함으로 인해, 복잡도가 30% 정도 증가하게 되었다.

<표 8>은 1비트 해상도 비트를 사용하는 방식에 해상도

표 7. 1비트 해상도 비트를 항상 이용하는 적응적 움직임벡터 해상도의 실험 결과
Table 7. Summary of test results for adaptive motion vector resolution using 1bit resolution flag

	Random Access Main			Random Access High Efficiency		
	BD-rate Y	BD-rate U	BD-rate V	BD-rate Y	BD-rate U	BD-rate V
Class A	0.3%	-0.4%	-0.4%	0.1%	-0.1%	-0.2%
Class B	-0.1%	-0.4%	-0.5%	0.0%	-0.2%	-0.4%
Class C	-1.0%	-1.1%	-1.0%	-0.5%	-0.9%	-0.8%
Class D	-1.4%	-1.3%	-1.7%	-0.6%	-1.1%	-1.6%
Enc Time[%]	134%			132%		
Dec Time[%]	102%			105%		

	Low delay B Main			Low delay B High Efficiency		
	BD-rate Y	BD-rate U	BD-rate V	BD-rate Y	BD-rate U	BD-rate V
Class B	0.1%	-0.3%	0.1%	0.1%	0.5%	1.1%
Class C	-1.0%	-1.3%	-1.5%	-0.5%	0.1%	-0.1%
Class D	-1.6%	-1.7%	-2.4%	-0.9%	0.2%	-0.8%
Enc Time[%]	133%			132%		
Dec Time[%]	103%			100%		

표 8. 1비트 해상도 비트 유추 방식을 이용한 방법의 실험 결과
Table 8. Summary of test results for adaptive motion vector resolution using 1-bit resolution flag inference method

	Random Access Main			Random Access HE10		
	BD-rate Y	BD-rate U	BD-rate V	BD-rate Y	BD-rate U	BD-rate V
Class A	0.3%	-0.5%	-0.4%	0.1%	-0.2%	-0.2%
Class B	-0.1%	-0.5%	-0.5%	0.0%	-0.2%	-0.4%
Class C	-1.0%	-1.1%	-1.0%	-0.5%	-1.0%	-0.8%
Class D	-1.5%	-1.3%	-1.8%	-0.6%	-1.1%	-1.7%
Enc Time[%]	134%			132%		
Dec Time[%]	103%			105%		

	Low delay B Main			Low delay B HE10		
	BD-rate Y	BD-rate U	BD-rate V	BD-rate Y	BD-rate U	BD-rate V
Class B	0.1%	-0.3%	0.1%	0.1%	0.4%	1.0%
Class C	-1.1%	-1.3%	-1.5%	-0.6%	0.1%	-0.1%
Class D	-1.7%	-1.7%	-2.5%	-1.0%	0.1%	-0.8%
Enc Time[%]	133%			131%		
Dec Time[%]	104%			101%		

플래그를 유추하는 방식을 적용하였다. MVP와 MVD가 둘다 짝수인 경우 해상도 비트를 디코더로 전송을 안하며, 그에 따라 cost가 변경되어 움직임 벡터가 변경될 수 있다. 이때 색차의 경우 움직임 벡터를 직접 추정하는 방식이 아닌 휘도의 움직임 벡터를 다운샘플링하여 사용하는 방식이므로 움직임 벡터의 변경으로 인한 성능의 차이가 생길 수 있다. 따라서 본 논문에서는 색차의 성능저하를 막기 위해 해상도 비트를 전송 안하는 경우에도 해상도 비트를 포함하여 기존과 동일하게 cost를 결정하고, 실제 부호화를 수행하는 경우에만 해상도 비트를 전송하지 않는 방식으로 결정하여 색차의 성능저하를 방지하였다.

<그림 5>은 <표 8>에서 사용된 방식인 해상도 비트 유추가 가능한, 움직임 벡터 예측값과 움직임 벡터 차이값이 모두 1/4 화소 위치인 경우와 아닌 경우를 그림으로 표현하였다. 모두 1/4 화소 위치인 경우 한 비트 절약 할 수 있기 때문에 전체적인 BD-rate가 개선될 수 있다. 빈도 수는 모두 1/4 화소 위치가 아닌 경우보다는 적지만 분명히 존재할 수 있는 통계적 특성이다. <표 9>~<표 12>는 <표 8>의 영상별 실험 결과이다.

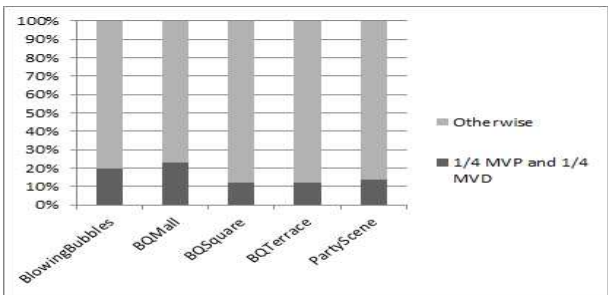

그림 5. MVP와 MVD 모두 1/4 위치인 경우의 비율
Fig. 5. Ratio of the case with 1/4 pixel position MVP and MVD

표 9. Random Access Main의 BD-rate 성능 결과
Table 9. BD-rate results for Random Access Main

Class	Sequence	BD-rate Y	BD-rate U	BD-rate V
A 4K	Traffic	0.2%	0.1%	-0.1%
	PeopleOnStreet	0.3%	0.5%	0.2%
	Nebuta	0.2%	-2.2%	-1.9%
	StreamLocomotive	0.4%	-0.2%	0.1%
B 1080p	Kimono	0.2%	0.2%	0.0%
	ParkScene	0.1%	-0.5%	-0.4%
	Cactus	-0.2%	-0.1%	-0.6%
	BasketballDrive	0.1%	0.0%	-0.3%
	BQTerrace	-0.6%	-1.9%	-1.2%
C WVGA	BasketballDrill	-1.0%	-1.4%	-1.2%
	BQMall	-1.0%	-0.5%	-0.8%
	PartyScene	-2.1%	-2.6%	-2.3%
	RaceHorses	0.1%	0.0%	0.1%
D WQVGA	BasketballPass	-0.1%	-0.1%	-0.7%
	BQSquare	-5.3%	-3.0%	-5.2%
	BlowingBubbles	-0.9%	-2.2%	-1.6%
	RaceHorses	0.3%	0.0%	0.4%

표 10. Random Access HE10의 BD-rate 성능 결과
Table 10. BD-rate results for Random Access HE10

Class	Sequence	BD-rate Y	BD-rate U	BD-rate V
A 4K	Traffic	0.3%	-0.1%	-0.2%
	PeopleOnStreet	0.4%	0.3%	0.2%
	Nebuta	0.2%	-0.4%	0.2%
	StreamLocomotive	-0.5%	-0.4%	-1.0%
B 1080p	Kimono	0.1%	0.4%	0.0%
	ParkScene	0.2%	-0.3%	-0.2%
	Cactus	0.0%	-0.2%	-0.3%
	BasketballDrive	0.1%	-0.1%	-0.4%
	BQTerrace	-0.3%	-0.9%	-1.0%
C WVGA	BasketballDrill	-0.5%	-1.0%	-0.9%
	BQMall	-0.7%	-0.7%	-0.5%
	PartyScene	-1.0%	-2.2%	-2.1%
	RaceHorses	0.0%	0.1%	0.1%
D WQVGA	BasketballPass	0.1%	-0.2%	-0.9%
	BQSquare	-2.4%	-3.2%	-4.8%
	BlowingBubbles	-0.4%	-1.0%	-0.8%
	RaceHorses	0.1%	-0.2%	-0.2%

표 11. Lowdelay B Main의 BD-rate 성능 결과
Table 11. BD-rate results for Lowdelay B Main

Class	Sequence	BD-rate Y	BD-rate U	BD-rate V
B 1080p	Kimono	0.2%	0.1%	0.4%
	ParkScene	0.2%	0.0%	-0.3%
	Cactus	-0.1%	-0.4%	0.3%
	BasketballDrive	0.0%	-1.2%	-0.8%
	BQTerrace	0.2%	0.0%	0.9%
C WVGA	BasketballDrill	-1.2%	-3.2%	-3.6%
	BQMall	-1.1%	-0.2%	-0.4%
	PartyScene	-2.1%	-1.9%	-2.2%
	RaceHorses	0.0%	0.1%	0.1%
D WQVGA	BasketballPass	-0.1%	-0.3%	0.2%
	BQSquare	-5.2%	-4.6%	-7.9%
	BlowingBubbles	-1.6%	-2.3%	-2.1%
	RaceHorses	0.1%	0.3%	-0.1%

표 12. Lowdelay B HE10의 BD-rate 성능 결과
Table 12. BD-rate results for Lowdelay B HE10

Class	Sequence	BD-rate Y	BD-rate U	BD-rate V
B 1080p	Kimono	0.0%	0.6%	0.4%
	ParkScene	0.2%	-0.1%	0.4%
	Cactus	-0.1%	-0.3%	-0.2%
	BasketballDrive	0.2%	0.3%	0.5%
	BQTerrace	0.4%	1.7%	4.1%
C WVGA	BasketballDrill	0.1%	0.1%	-0.5%
	BQMall	-1.0%	0.3%	0.1%
	PartyScene	-1.4%	-0.2%	-0.1%
	RaceHorses	0.0%	0.1%	0.0%
D WQVGA	BasketballPass	0.0%	-0.3%	-0.6%
	BQSquare	-2.7%	0.8%	-2.5%
	BlowingBubbles	-1.0%	0.3%	0.1%
	RaceHorses	-0.1%	-0.3%	-0.2%

3. 결과 분석

<표 13>~<표 15>는 <표 9>에서 성능이 좋게 나온 영상과 나쁘게 나온 영상을 뽑아 분석을 한 결과이다.

표 13 Random Access Main의 B_class BQSquare 영상 분석
Table 13 The evaluation about B_class BQSquare sequence at Random Access

D_BQSquare					
QP	1/4 MV	1/8 MV	△Y (dB)	△Bitrate (%)	BD-rate
22	41%	59%	0.11 dB	-4.23 %	-5.3%
27	38%	62%	0.16 dB	-2.42 %	
32	36%	64%	0.12 dB	-1.65 %	
37	37%	63%	0.09 dB	-0.57 %	
average	38%	62%	0.12 dB	-2.22 %	

표 14. Random Access Main의 C_class RaceHorses 영상 분석
Table 14. The evaluation about C_class RaceHorses sequence at Random Access

C_RaceHorses					
QP	1/4 MV	1/8 MV	△Y (dB)	△Bitrate (%)	BD-rate
22	48%	52%	0.00 dB	0.01 %	0.1%
27	51%	49%	0.01 dB	0.18 %	
32	55%	45%	-0.01 dB	0.06 %	
37	56%	44%	0.00 dB	0.26 %	
average	53%	47%	0.00 dB	0.13 %	

표 15. Random Access Main의 A_class SteamLocomotiveTrain 영상 분석
Table 15. The evaluation about A_class SteamLocomotiveTrain sequence at Random Access

A_SteamLocomotiveTrain					
QP	1/4 MV	1/8 MV	△Y (dB)	△Bitrate (%)	BD-rate
22	65%	35%	0.00 dB	-0.13 %	0.4%
27	58%	42%	0.00 dB	0.17 %	
32	58%	42%	-0.01 dB	0.41 %	
37	57%	43%	0.00 dB	0.07 %	
average	60%	40%	0.00 dB	0.13 %	

위의 표에서 1/4 MV와 1/8 MV 는 MV 해상도의 선택된 비율을 나타내고 있으며, △Y는 dB의 변화량, △Bitrate는

비트의 변화량을 나타내고 있다.

HM 6.0 대비 -5.3%의 BD-rate 향상을 보이는 BQSquare의 경우, 1/8 해상도의 비율이 평균 60% 이상이며 그에 따른 dB도 평균 0.12dB 상승을 하였고, 비트량도 평균 2% 줄었다. 따라서 BQSquare의 경우 1/8 해상도를 이용하는 방식이 최적임을 알 수 있다. RaceHorses의 경우 1/4 해상도의 선택 비율이 1/8 해상도보다 높으며, 평균 dB의 변화량이 0인데 반해 비트량은 평균 0.13% 증가함을 볼 수 있다. 따라서 이 경우, 오버헤드를 감당하지 못하여 손해가 발생하는 것을 알 수 있다. 성능이 가장 떨어지는 Stream LocomotiveTrain의 경우 1/8 해상도의 비율이 평균 40%에 불과하며 dB의 변화는 없는데 반해 비트량이 증가하였다. 이 영상도 RaceHorses와 마찬가지로 오버헤드를 감당하지 못하여 손해가 발생하는 것을 알 수 있다. 위의 세 영상 이외에 다른 영상들도 마찬가지로, 1/4 해상도와 1/8 해상도의 비율에 따라 성능의 차이가 발생하는 것을 알 수 있다.

V. 결 론

본 논문에서는 1/4 화소 위치까지 움직임 예측을 하는 기존의 HEVC 방식과는 달리, 1/8 화소 위치까지 움직임 벡터를 찾은 후, 1비트 해상도 플래그를 두어 적응적으로 움직임 벡터의 해상도를 결정하는 방식을 제안 하였다. 제안한 방법을 HEVC에 적용한 결과 부호화 시간은 약 30%정도, 복호화 시간은 5% 이내로 증가하였다. 휘도의 압축효율은 영상 별로 최대 5.3% 좋아졌으며, 색차신호의 압축효율은 최대 7.9% 좋아졌다.

참 고 문 헌

[1] ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 Part 10 AVC), "Advanced Video Coding for Generic Audiovisual Services," Nov. 2007.

[2] Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, "High efficiency video coding(HEVC) text specification draft 6," JCTVC-H1003, San Jose, USA, Feb. 2012.

[3] Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, "Video coding technology proposal by Samsung(and BBC)," JCTVC-A124, Dresden, DE, Apr. 2010.

[4] ITU-T Telecommunications Standardization Sector, "1/8-Pel motion vector resolution for H.26L," Q15-K-21, Portland, Oregon, USA, Aug. 2000.

[5] J. Ribas-Corbera and D. L. Neuhoff, “Optimizing motion-vector accuracy in block-based video coding,” IEEE Trans. Circuits Syst. Video Technol., vol. 11, Apr. 2001.

[6] Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, "CE3:7 taps interpolation filters for quarter pel position MC from Samsung and Motorola Mobility," JCTVC-G778, Geneva, CH, Nov. 2011.

[7] Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, "CE3:Progressive Motion Vector Resolution(Tool10)," JCTVC-G277, Geneva, CH, Nov. 2011.

[8] Http://hevc.kw.bbc.co.uk/trac/browser/tags/HM-6.0

[9] G. Bj “Caculation of average PSNR differences between RD curves," in ITU-T SC16/Q6 13th VCEG meeting, No. VCEG-M33, Austin

[10] Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, "Common test conditions and software reference configurations," JCTVC-H1100, San Jose, USA, Feb. 2012.

저 자 소 개

임 성 원

- 2011년 : 세종대학교 정보통신공학과 공학사
- 2011년 ~ 현재 : 세종대학교 대학원 정보통신공학과 석사과정
- 주관심분야 : Data compression, Image coding & transmission

저 자 소 개

이 주 옥

- 2009년 : 세종대학교 정보통신공학과 공학사
- 2011년 : 세종대학교 대학원 정보통신공학과 공학석사
- 2011년 ~ 현재 : 세종대학교 대학원 정보통신공학과 박사과정
- 주관심분야 : Data compression, Image coding & transmission, Digital TV, MPEG, Medical Imaging

문 주 희

- 1985년 : 서강대학교 전자공학과 학사
- 1987년 : 한국과학기술원 전기및전자공학과 석사
- 1992년 : 한국과학기술원 전기및전자공학과 박사
- 1992년 ~ 1994년 : 한국과학기술원 전자정보연구소
- 1994년 ~ 1999년 : 현대전자 정보통신연구소 수석연구원
- 1999년 ~ 현재 : 세종대학교 정보통신공학과 교수
- 주관심분야 : Data compression, Image coding & transmission, Digital TV, MPEG, Medical Imaging