2014년 12월 전자공학회 논문지 제 51권 제 12 호
Journal of The Institute of Electronics and Information Engineers Vol. 51, NO. 12, December 2014 http://dx.doi.org/10.5573/ieie.2014.51.12.150

논문 2014-51-12-16

HEVC 부호화기를 위한 고속 비정수 움직임 추정

(Fast Non-integer Motion Estimation for HEVC Encoder)

한 우 진*

(Woo-Jin Han©)

요 약

최신 영상 압축 표준 방식인 HEVC는 H.264/AVC에 비해 압축 효율을 크게 개선시킬 수 있지만, 부호화기 복잡도 또한 크게 증가한다. 특히 비정수 정밀도 움직임 보상에 사용되는 보간 필터의 길이가 종래 6-tap에서 8-tap으로 증가함으로 인해, 비정수 정밀도 움직임 추정에 많은 연산량이 요구된다. 본 논문에서는 HEVC의 비정수 움직임 추정 과정에 대한 압축 효율 기여도 및 복잡도를 분석하고, 이로부터 부호화기의 복잡도를 효과적으로 감소시키기 위한 방법을 제안한다. 먼저, 움직임 추정과 움직임 보상에 사용되는 보간 필터를 분리하고, 움직임 추정만을 위한 최적 필터 길이를 찾는다. 또한 최적 비정수 움직임 벡터를 찾기 위한 탐색 과정에서 특정 조건을 만족하는 일부 후보들만을 검사하고, 꼭 필요한 보간 과정만을 수행하도록 함으로써 부호화 복잡도를 감소시킨다. 실험 결과, 제안한 방법을 사용하면 평균 압축 성능 하락 폭 0.7%, 1.5%, 2.5%에서 부호화기 복잡도를 각각 13.6%, 18.5%, 21.1% 감소시킬 수 있었다. 또한 고해상도 영상(1920x1080)의 경우 압축 성능 하락 폭이 0.4%, 1.1%, 1.6%로 감소함으로써 제안한 방법이 고해상도 영상에 더욱 효과적임을 보였다.

Abstract

The latest video coding standard, HEVC can improve the coding efficiency significantly compared with the H.264/AVC. However the HEVC encoder requires much larger computational complexities. The longer 8-tap interpolation filter of the HEVC which is used in a non-integer motion estimation is one of the reasons and this paper aims to reduce the computational complexities. First of all, three shorter-tap interpolation filters for a motion estimation process are tested rather than the use of a standard interpolation filter. In addition, the fast searching strategies to reduce the number of comparisons for choosing the best non-integer motion vector are proposed. Finally, the interpolation process is selectively applied according to the searching strategy. By combining all of the techniques, the experimental results show that the encoding times can be reduced by 13.6%, 18.5% and 21.1% with the coding efficiency penalties of 0.7%, 1.5% and 2.5%, respectively. For the full-HD video sequences, the coding efficiency penalties are reduced to 0.4%, 1.1% and 1.6% at the same level of the encoding time savings, which shows the effectiveness of the proposed schemes for the high resolution video sequences.

Keywords : HEVC, encoding, fast-algorithm, motion estimation

Ⅰ. 서 론

최근 표준화가 완료된 HEVC[1]는 H.264/AVC[2] 대비 압축 효율을 약 40% 이상 향상시킬 수 있는 것으로 알려져 있다[3~4]. 그러나 이에 따른 부호화기 및 복호화기의 복잡도 또한 크게 증가하여, 이를 감소시키기 위한 많은 연구들이 진행되어 왔다.

HEVC에서는 항상 16x16 크기를 갖는 매크로블록(macroblock)으로 화면을 분할하는 H.264/AVC와는 달리 최대 64x64까지의 크기를 갖는 coding tree unit (CTU)들로 화면을 분할한 후, 각 CTU들을 쿼드트리 형태로 재귀적 분할하여 최소 8x8 크기를 갖는 coding unit (CU)들로 나누는 구조를 가지고 있다[5]. 이러한 자

* 정회원, 가천대학교 소프트웨어학과
(Dept. of Software, Gachon University)
© Corresponding Author(E-mail: wjhan@gachon.ac.kr)
접수일자: 2014년11월20일, 수정일자: 2014년11월27일,
게재확정: 2014년12월01일

유로운 분할 형태 때문에 HEVC의 부호화기에서는 최적 분할 구조를 빠르게 찾는 것이 중요하며, 이를 위한 많은 고속화 연구들이 발표되었다[6~9].

기존 연구의 접근 방법들이 부호화기의 복잡도를 크게 감소시키면서도 부호화 효율을 유지할 수 있는 것은 사실이나, 최적 분할 구조 결정이라는 유사한 목적을 갖고 있기 때문에 다수의 고속화 기법들을 동시에 적용하였을 경우 추가적인 복잡도 향상에는 한계가 있다. 따라서 추가적인 복잡도 감소를 위해서는 최적 분할 구조 결정과는 독립적인 부분에서의 복잡도 감소가 필요하다.

HEVC의 화면 내 예측의 경우 종래 H.264/AVC의 최대 9가지 모드에서 35가지 모드로 확장됨으로 인하여 부호화 복잡도가 크게 증가하였다. 이를 단순화시키기 위한 연구로서, [10]에서는 gradient를 미리 계산하여 탐색 후보 방향을 제한하였고, [11]에서는 주변에 위치한 이미 부호화된 화면 내 예측 블록의 기 결정된 방향 정보를 활용하여 복잡도를 크게 감소시켰다. 그러나 일반적으로 부호화기에서 화면 간 예측이 차지하는 복잡도가 화면 내 예측에 비해서 월등히 높기 때문에 화면 간 예측 부호화의 복잡도를 감소시키는 것은 매우 중요하다.

HEVC에서는 H.264/AVC와 마찬가지로 1/4 정밀도를 갖는 움직임 벡터를 허용하지만, 비정수 움직임 보상을 수행하기 위해 필요한 보간 과정에 사용되는 필터의 길이가 6-tap에서 8-tap으로 증가하였고, 이로 인해 비정수 움직임 추정 단계의 복잡도 또한 크게 증가하였다. [12~13]에서는 비정수 움직임 추정 단계의 복잡도를 감소시키기 위해 주변 정수 위치에서의 에러를 반영하였고, 복잡도를 크게 감소시켰지만, 고속 정수 움직임 추정 알고리즘이 함께 사용되는 경우에 대한 고려가 없으며, HEVC에서 변화된 보간 과정을 반영하지 않았다.

본 논문에서는 HEVC 부호화기의 비정수 움직임 추정 단계가 압축 효율에 기여하는 정도 및 복잡도를 분석하였고, 이 결과로부터 화면 간 예측의 복잡도를 감소시킬 수 있는 방법을 고안하였다. 구체적으로, 보간 필터 적용의 복잡도를 감소시키기 위하여 움직임 추정 단계와 움직임 보상 단계에서 상이한 보간 필터를 사용하였으며, 보간 후 최적 비정수 움직임 벡터 탐색 과정에서도 전체 후보 위치 대신 일부분만을 탐색함으로써 복잡도를 감소시켰다. 한편, 독립적으로 사용되었을 경우는 물론, 최적 분할 구조를 얻기 위해 사용되는 고속 알고리즘과 함께 적용하였을 경우에 대해서도 추가적인 속도 향상의 결과를 보인다.

본 논문의 구성은 다음과 같다. 제 Ⅱ장에서는 HEVC 부호화기의 비정수 움직임 추정 과정을 개략적으로 소개한 후 압축 효율 및 복잡도에 대해서 분석한 결과를 보인다. 제 Ⅲ장에서는 이 결과로부터 복잡도를 감소시키기 위한 알고리즘을 제안하고, 제 Ⅳ에서 성능 평가 결과를 보인 후, 마지막으로 제 Ⅴ장에서 결론을 맺는다.

Ⅱ. HEVC 부호화기의 비정수 움직임 추정 과정

HEVC 부호화기에서는 정수 정밀도 최적 움직임 벡터를 결정한 후, 그 위치를 기준으로 1/2 정밀도를 갖는 주위 9개 지점에 대해서 최적 1/2 정밀도 움직임 벡터를 탐색한다. 그 후, 결정된 지점을 기준으로 다시 1/4 정밀도를 갖는 주위 9개 지점에 대해서 최종적으로 최적 1/4 정밀도 움직임 벡터를 결정한다. 한편, 1/2 정밀도 및 1/4 정밀도 움직임 벡터 탐색 과정 각각에 대해, 9개 후보 지점에 대한 보간 블록 생성 및 최적 비정수 움직임 벡터 탐색 단계가 필요하다.

1. 9개 후보 지점에 대한 보간 블록 생성

주어진 비정수 움직임 벡터에 대응되는 보간 블록 하나만을 생성하는 움직임 보상 단계와는 달리, 움직임 추정 단계에서는 1/2 정밀도 및 1/4 정밀도 각각에 대해 9개 지점의 보간 블록을 생성해야 한다. 1/2 정밀도 및 1/4 정밀도를 갖는 샘플을 얻기 위해서는 정수 정밀도 샘플들에 보간 필터를 적용하게 되는데, HEVC에서는 휘도 성분에 대해 8-tap, 색차 성분에 대해 4-tap DCT-IF[14]를 사용한다.

정수 정밀도 움직임 벡터 (x, y)가 주어졌을 때, 가로, 세로 방향으로 각각 1/2 정밀도만큼의 차이를 갖는 9개 후보 지점을 각각 (x-0.5, y-0.5), (x, y-0.5), (x+0.5, y-0.5), (x-0.5, y), (x, y), (x+0.5, y), (x-0.5, y+0.5), (x, y+0.5), (x+0.5, y+0.5)라고 하자. 이 때, (x, y) 지점은 정수 정밀도 지점이므로 별도로 보간 필터를 적용할 필요가 없으며, (x-0.5, y)와 (x+0.5, y)는 가로 방향 차이가 정수 샘플 단위이므로 하나의 블록으로 표현 가능하다. 마찬가지로 (x, y-0.5)와 (x, y+0.5), 그리고 (x-0.5,

y-0.5), (x+0.5, y-0.5), (x-0.5, y+0.5), (x+0.5, y+0.5)는 각각 하나의 블록으로 표현된다. 그림 1은 이를 나타낸 것이다.

한편, 이를 이용한 1/2 정밀도 보간 블록 생성 과정은 다음과 같다.

- Step #1: 정수 정밀도 샘플들을 블록 A에 저장한다.
- Step #2: 정수 정밀도 샘플에 대해 가로 방향으로 1/2 정밀도 보간 필터를 적용하여 (x-0.5, y)와 (x+0.5, y) 지점의 예측 샘플들을 모두 포함하는 보간 블록 B를 얻는다.
- Step #3: 정수 정밀도 샘플에 대해 세로 방향으로 1/2 정밀도 보간 필터를 적용하여 (x, y-0.5)와 (x, y+0.5) 지점의 예측 샘플들을 모두 포함하는 보간 블록 C를 얻는다.
- Step #4: 블록 B에 세로 방향으로 1/2 정밀도 보간 필터를 다시 적용하여 (x-0.5, y-0.5), (x+0.5, y-0.5), (x-0.5, y+0.5), (x+0.5, y+0.5) 지점의 예측 샘플들을 모두 포함하는 보간 블록 D를 얻는다.

위의 과정에서 볼 수 있듯이, 1/2 정밀도를 갖는 후보 블록 생성을 위해서는 Step #2, Step #3, Step #4, 총 3번의 보간 과정이 필요하다.

한편, 1/2 정밀도 최적 움직임 벡터 (x, y)가 주어졌

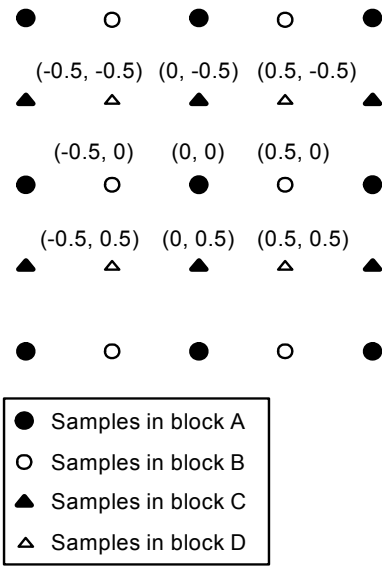

그림 1. 1/2 정밀도의 각 후보 지점 및 대응되는 보간 블록들

Fig. 1. Candidate half-pel positions and corresponding interpolation blocks.

을 때, 가로, 세로 방향으로 각각 1/4 정밀도만큼의 차이를 갖는 9개 후보 지점을 각각 (x-0.25, y-0.25), (x, y-0.25), (x+0.25, y-0.25), (x-0.25, y), (x, y), (x+0.25, y), (x-0.25, y+0.25), (x, y+0.25), (x+0.25, y+0.25)라고 하자. 이에 대한 보간 블록 생성 과정은 다음과 같다.

- Step #1: 1/2 정밀도 샘플들을 보간 블록 A에 저장한다.
- Step #2: 정수 정밀도 샘플에 대해 가로 방향으로 1/4 정밀도, 3/4 정밀도 보간 필터를 각각 적용하여 (x+0.25, y), (x-0.25, y) 지점의 보간 블록 B, C를 얻는다.
- Step #3: 블록 A에 세로 방향으로 1/4 정밀도, 3/4 정밀도 보간 필터를 각각 적용하여 (x, y+0.25), (x, y-0.25) 지점의 보간 블록 D, E를 얻는다.
- Step #4: 블록 B에 세로 방향으로 1/4 정밀도, 3/4 정밀도 보간 필터를 각각 적용하여 (x+0.25, y+0.25), (x+0.25, y-0.25) 지점의 보간 블록 F, G를 얻는다.
- Step #5: 블록 C에 세로 방향으로 1/4 정밀도, 3/4 정밀도 보간 필터를 각각 적용하여 (x-0.25, y+0.25), (x-0.25, y-0.25) 지점의 보간 블록 H, I를 얻는다.

1/4 정밀도의 경우 1/4 정밀도, 3/4 정밀도의 두 가지 보간 필터가 존재하며 각 샘플간 거리가 정수 샘플 단위가 아니므로 1/2 정밀도와는 달리 각각의 지점들을 모두 별도의 보간 과정을 통해 생성해야 한다. 따라서 1/4 정밀도의 총 보간 횟수는 8번이며, 1/2 정밀도에 비해 2배 이상의 더 많은 연산량을 요구한다.

2. 최적 비정수 움직임 벡터 탐색

보간 블록들에 대한 생성이 완료되면, 이들로부터 9개의 후보 지점에 각각 대응되는 예측 블록들을 만들 수 있다. 블록의 크기를 NxM, (x+a, y+b) 지점에 대한 예측 블록을 $P_{a,b}$, 원본 블록을 O라고 하면 각 예측 블록들에 대한 율-왜곡 값 (rate-distortion cost) Ra,b는 식 (1)과 같이 구할 수 있다.

$$R_{a,b} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} |O(x,y) - P_{a,b}(x,y)| + \lambda B_{a,b} \qquad (1)$$

단, 식 (1)에서 λ는 Lagrangian multiplier이며 $B_{a,b}$는 움직임 벡터를 부호화하기 위해서 필요한 비트수이다.

1/2 정밀도에 대해서는 a, b를 각각 -0.5, 0, +0.5로, 1/4 정밀도에 대해서는 a, b를 각각 -0.25, 0, +0.25로 변화시키면서 식 (1)을 계산하고, 그 중 최소값을 갖는 a와 b의 값을 취함으로써 최적 비정수 움직임 벡터 값을 확정한다.

3. 압축 효율 및 복잡도 분석

HEVC의 비정수 움직임 벡터 추정 과정은 크게 1/2 정밀도 보간 블록 생성, 1/2 정밀도 최적 벡터 탐색, 1/4 정밀도 보간 블록 생성, 1/4 정밀도 최적 벡터 탐색의 네 가지 단계로 이루어져 있다. 각각의 단계에 대한 압축 효율 및 복잡도를 분석하기 위해서 HEVC의 참조 소프트웨어인 HM15.0[15]에 기반한 다음의 시스템들을 구현하고, 그 성능 및 부호화 시간을 측정하였다.

- S1: HM15.0에서 비정수 움직임 벡터 추정 제거
- S2: S1 + 1/2단위 보간 블록 생성
- S3: S2 + 1/2단위 최적 벡터 탐색
- S4: S3 + 1/4단위 보간 블록 생성

테스트를 위해서는, HEVC 참조 소프트웨어의 성능을 평가하기 위해 표준화 기구인 JCT-VC에서 사용하는 common test condition[16] 중, 부호화 지연시간 제한에 의한 성능 저하가 없어 가장 높은 성능을 보이는 random-access configuration을 사용하였다. 또한, [16]에 정의된 영상 중 Class D(416x240) 4개 영상, Class C(832x480) 4개 영상, Class B(1920x1080) 5개 영상의 총 13개 영상이 테스트에 사용되었다.

표 1은 S1-S4 각각의 경우에 대해, HM15.0 대비 BD-rates[17] 및 상대적인 부호화 시간을 나타낸 것이다. 상대적인 부호화 시간은 식 (2)와 같이, 평가하고자 하는 시스템의 부호화 시간 Time(Proposed)를 HM15.0의 부호화 시간 Time(Reference)로 나눈 백분율로 측정하였다. 따라서 BD-rates 상의 양수 값은 압축 효율이 하락했음을 의미하며, 상대적인 부호화 시간이 100보다 작으면 복잡도가 HM15.0 대비 감소했음을 의미한다.

$$Rel.enc.time = \frac{Time(Proposed)}{Time(Reference)} \quad (2)$$

표 1. S1-S4에 대한 압축 효율 및 부호화 시간
Table 1. Coding efficiency and relative encoding time of S1-S4.

System	BD-rates (%)				Rel. enc. time (%)
	Class B	Class C	Class D	Average	
S1	11.4	14.4	16.6	14.4	61.9
S2	11.4	14.4	16.6	14.4	71.0
S3	3.1	4.2	5.2	4.1	77.9
S4	3.1	4.2	5.2	4.1	93.1

표 1의 결과로부터 비정수 움직임 벡터 추정의 각 단계들에 대한 압축 효율 및 복잡도에 대한 영향들을 분석할 수 있다. 먼저, S1의 경우 평균 14.4%의 압축 효율 하락과 38.1%의 부호화 복잡도 감소를 보였는데, 이는 비정수 움직임 벡터 추정 전체가 갖는 압축 효율 및 복잡도에 대한 영향을 의미한다.

S2의 경우 S1에 1/2단위 보간 블록 생성 과정만을 추가한 것이기 때문에 압축 효율에는 변화가 없지만 부호화 시간이 61.9%에서 71.0%로 증가함으로써 1/2단위 보간 블록 생성 과정이 약 9.1% 가량의 복잡도를 증가시킴을 알 수 있다. S3의 경우 S2에 1/2단위 최적 벡터 탐색을 추가한 것이며, 압축 효율 감소가 14.4%에서 4.1%로 감소하였고, 6.9%의 복잡도가 증가되었다.

S4는 S3에 1/4단위 보간 블록 생성 과정을 추가한 것이며, 부호화 시간이 77.9%에서 93.1%로 크게 증가함으로써 이 과정이 약 15.2%의 복잡도를 증가시킴을 알 수 있다. 표 2는 위의 분석 결과들을 요약 정리한 것이다. 표 2의 결과로부터, 1/4 보간 블록 생성 단계가 가장 많은 부호화 시간 증가를 가져옴을 알 수 있으며, 제Ⅲ장에서는 보간 블록 생성 및 최적 벡터 탐색 단계 각각에 대한 복잡도 감소 방안을 제안한다.

표 2. 비정수 움직임 벡터 추정 과정 각 단계의 압축 효율 및 부호화 시간에 미치는 영향
Table 2. Coding efficiency and relative encoding time of each part of non-integer motion estimation.

Module	Coding efficiency	Rel. enc. time (%)
1/2 보간 블록 생성	10.3	9.1
1/2 최적 벡터 탐색		6.9
1/4 보간 블록 생성	4.1	15.2
1/4 최적 벡터 탐색		6.9

Ⅲ. 비정수 움직임 추정 고속화

1. 움직임 추정을 위한 보간 필터 길이 최적화

일반적으로 최적의 압축 성능을 보장하기 위해서, 움직임 추정을 위해 사용하는 보간 필터는 움직임 보상을 위한 보간 필터와 동일한 것을 사용한다. 하지만 HEVC에서는 8-tap DCT-IF를 보간 필터로 사용하기 때문에 보간에 필요한 복잡도가 상대적으로 높고, 이는 표 2에서 볼 수 있듯이 1/2 보간 블록 생성과 1/4 보간 블록 생성을 합쳐 약 24.3%의 부호화기 복잡도 상승을 가져온다.

본 논문에서는 움직임 추정을 위한 보간 필터와 움직임 보상을 위한 보간 필터를 서로 분리하고, 움직임 추정을 위한 보간 필터의 탭 수를 변화시켜가면서 압축 성능과 복잡도에 미치는 영향을 분석함으로써 압축 효율과 복잡도 면에서 최적인 보간 필터를 선택하고자 한다. 이를 위해 필터 탭 수 2를 갖는 bi-linear 필터, 필터 탭 수 4와 6을 갖는 DCT-IF 필터[14]들을 각각 생성한 뒤, 각각 HM15.0의 움직임 추정부에 구현하였으며 각 필터들의 계수들은 표 3에 정리하였다. 또한, 표 3의 필터 계수들은 HEVC 규격에 정의되어 있는 8-tap 필터와 동일한 정밀도를 갖도록 필터 계수의 합이 64가 되도록 조정하였다.

표 3. 움직임 추정에서의 보간 필터 길이 최적화를 위한 필터 계수들
Table 3. Filter coefficients for optimizing the number of filter taps used in motion estimation.

Number of filter taps	Non-integer position	Filter coefficients
2	1/4	{ 48, 16 }
	1/2	{ 32, 32 }
	3/4	{ 16, 48 }
4	1/4	{ -6, 56, 18, -4 }
	1/2	{ -8, 40, 40, -8 }
	3/4	{ -4, 18, 56, -6 }
6	1/4	{ 2, -8, 56, 18, -6, 2 }
	1/2	{ 2, -10, 40, 40, -10, 2 }
	3/4	{ 2, -6, 18, 56, -8, 2 }

2. 탐색 횟수를 감소시키기 위한 고속 탐색 전략

1/2 정밀도, 1/4정밀도 모두 최적의 비정수 움직임 벡터를 추정하기 위해서는 제 Ⅱ장에서 설명한 바와 같이 총 9번의 비교 과정을 거친다. 그러나 일반적으로 정수 정밀도 움직임 추정 과정에서 사용되는 고속 알고리즘들은 특정한 조건을 만족하는 경우에 대해서만 탐색을 수행하도록 함으로써 총 탐색 횟수를 감소시키는데, 대표적인 알고리즘으로 탐색 범위 중심점을 기준으로 diamond 형태의 패턴을 먼저 탐색하고, 그 결과에 따라서 나머지 부분을 선택적으로 탐색하는 diamond search[18]가 있다.

본 논문에서는 정수 정밀도 움직임 추정 과정에서 사용되는 diamond search 및 변형된 방법들을 비정수 움직임 벡터 탐색에 적용하고자 한다.

가. Diamond search

1/2 정밀도의 9개 후보 블록을 $P_{a,b}$ (a, b는 각각 -0.5, 0, +0.5)라고 하자. 1/4 정밀도의 경우도 유사하게 적용 가능하므로 이후에서는 1/2 정밀도에 대해서만 설명한다. 이 때, 9개의 후보 블록에 대한 rate-distortion

```
Function diamond_search( )
{
  Compute R0,0, R-0.5,0, R0,-0.5, R0.5,0, R0,0.5
  Rmin = Min( R0,0, R-0.5,0, R0,-0.5, R0.5,0, R0,0.5 )
  If R0,0 == Rmin
    Return (0, 0)
  Else
    If R-0.5,0 == Rmin
      Compute R-0.5,-0.5 and R-0.5,0.5
      Return best position among R-0.5,y
      (y=-0.5, 0, 0.5)
    Else If R0,-0.5 == Rmin
      Compute R-0.5,-0.5 and R0.5,-0.5
      Return best position among Rx,-0.5
      (x=-0.5, 0, 0.5)
    Else If R0.5,0 == Rmin
      Compute R0.5,-0.5 and R0.5,0.5
      Return best position among R0.5,y
      (y=-0.5, 0, 0.5)
    Else If R0,0.5 == Rmin
      Compute R-0.5,0.5 and R0.5,0.5
      Return best position among Rx,0.5
      (x=-0.5, 0, 0.5)
}
```

그림 2. Diamond search의 슈도코드
Fig. 2. Pseudo-code of diamond search.

cost들을 모두 구하는 대신, $P_{0,0}$을 중심으로 한 diamond pattern인 $P_{0,0}$, $P_{-0.5,0}$, $P_{0,-0.5}$, $P_{0.5,0}$, $P_{0,0.5}$에 대한 rate-distortion cost $R_{0,0}$, $R_{-0.5,0}$, $R_{0,-0.5}$, $R_{0.5,0}$, $R_{0,0.5}$를 식 (1)에 의해 구한다. 그 결과, 중심 위치에 해당하는 $R_{0,0}$이 최소값을 갖는다면 (0, 0)을 최적 값으로 결정하며, 그렇지 않은 경우 아직 탐색하지 않은 후보 블록들 중 최적 위치와 인접한 2개를 추가로 탐색함으로써 최적 값을 결정한다. 그림 2는 이 과정에 대한 pseudo-code이다. Diamond search에서는 중심 위치 rate-distortion cost의 최소값 여부에 따라 총 탐색 횟수가 5회, 혹은 9회가 된다.

나. Band search

Band search에서는 먼저 중심 위치를 포함한 가로 방향 3개의 후보들에 대해서 탐색을 하고, 그 결과에 따라서 추가적으로 세로 방향 2개에 대한 후보들에 대한 탐색을 수행한다. 즉, diamond search에서는 첫 번째 단계에서 가로와 세로 모두를 고려하는데 비해, band search에서는 먼저 가로 방향을 고려하여 최적 위치를 선택한 후, 그 위치에 대해 세로 방향으로 최적 위치를 재탐색한다. 또한, diamond search에서는 첫 번째 단계에서 중심 위치가 최소값을 갖는 경우 더 이상의 탐색을 하지 않지만, band search에서는 이 경우에 대해서도 세로 방향으로 최적 위치를 재탐색한다. 그림 3은 band search에 대한 pseudo-code이다. Band search에서는 중심 위치의 rate-distortion cost에 관계없이 항상 총 탐색 횟수가 5회가 되므로, 9개를 모두 탐색하는 경우에 비해서 약 45% 가량 탐색 횟수를 감소시킬 수 있는 반면, diamond search에 비해서 압축 효율 하락 폭이 클 것으로 예상할 수 있다.

```
Function band_search( )
{
  Compute R0,0, R-0.5,0, R0.5,0
  Rmin = Min( R0,0, R-0.5,0, R0.5,0 )
  If R0,0 == Rmin
    Compute R0,-0.5 and R0,0.5
    Return best position among R0,y
    (y=-0.5, 0, 0.5)
  Else If R-0.5,0 == Rmin
    Compute R-0.5,-0.5 and R-0.5,0.5
    Return best position among R-0.5,y
    (y=-0.5, 0, 0.5)
  Else If R0.5,0 == Rmin
    Compute R0.5,-0.5 and R0.5,0.5
    Return best position among R0.5,y
    (y=-0.5, 0, 0.5)
}
```

그림 3. Band search의 슈도코드
Fig. 3. Pseudo-code of band search.

3. 고속 탐색 전략 기반 선택적 보간 블록 생성

앞 절에서 제안한 diamond search 혹은 band search를 사용하는 경우, 모든 후보 위치에 대한 rate-distortion cost를 계산하는 대신 조건에 따라 일부 후보의 rate-distortion cost만을 계산하게 된다. 따라서 계산에 필요한 보간 블록만을 생성함으로써 보간에 필요한 복잡도를 함께 감소시키고자 한다.

1/2 정밀도 보간 단계에서는 제 Ⅱ장에서 설명한 바와 같이 4개의 보간 블록 A, B, C, D를 생성하는데, diamond search를 사용하는 경우 보간 블록 A, B, C는

```
Function band_search_half( )
{
  Interpolate block A and block B
  Compute R0,0, R-0.5,0, R0.5,0
  Rmin = Min( R0,0, R-0.5,0, R0.5,0 )
  If R0,0 == Rmin
    Interpolate block C
    Compute R0,-0.5 and R0,0.5
    Return best position among R0,y
    (y=-0.5, 0, 0.5)
  Else If R-0.5,0 == Rmin
    Interpolate block D
    Compute R-0.5,-0.5 and R-0.5,0.5
    Return best position among R-0.5,y
    (y=-0.5, 0, 0.5)
  Else If R0.5,0 == Rmin
    Interpolate block D
    Compute R0.5,-0.5 and R0.5,0.5
    Return best position among R0.5,y
    (y=-0.5, 0, 0.5)
}
```

그림 4. 1/2 정밀도 보간을 위한 band search의 슈도코드
Fig. 4. Pseudo-code of band search for half accuracy interpolation.

첫 번째 단계에서 필요하지만 보간 블록 D는 필요하지 않다. 따라서 중심 위치가 최소값인 경우, 즉 첫 번째 단계만으로 탐색이 종료될 경우 보간 블록 D를 생성하는 과정을 생략함으로써 복잡도를 낮출 수 있다.

또한, band search의 경우에는 첫 번째 가로 탐색 단계에서 보간 블록 A, B 만이 필요하며, 두 번째 세로 탐색 단계에서는 첫 번째 단계의 결과에 따라 보간 블록 C 혹은 D가 필요하다. 이를 반영한 제안하는 1/2 정밀도 보간을 위한 band search의 pseudo-code는 그림 4와 같다. 이러한 방식으로 band search를 사용하면 1/2 정밀도의 경우 실질적으로 보간 과정이 필요 없는 보간 블록 A를 제외한다면 총 보간 횟수를 3회에서 2회로 약 33% 감소시킬 수 있다.

한편, 1/4 정밀도의 경우에도 유사하게 적용할 수 있는데, 이 경우에는 제 Ⅱ장에서 설명한 총 9개의 보간 블록 A, B, C, D, E, F, G, H, I 중, 첫 번째 가로 탐색 단계에서는 보간 블록 A, B, C를 계산해야 하며, 그 결과에 따라서 중심 위치가 최소값을 갖는 경우 보간 블록 D, E를, 왼쪽 위치가 최소값을 갖는 경우 보간 블록 F, G를, 오른쪽 위치가 최소값을 갖는 경우 H, I를 계산하면 된다. 이 경우 보간 블록 A를 제외한다면 총 보간 횟수를 8회에서 4회로 약 50% 감소시킬 수 있다.

Ⅳ. 실험 결과

1. 테스트 조건

제안한 방법의 효과를 검증하기 위해, HM15.0에 제안한 방법을 구현하고, 압축 효율 및 연산량 측면에서 비교를 수행하였다. 또한 HM15.0에 이미 구현되어 있는 early skip decision(ESD) 고속 탐색 알고리즘과의 상호 작용을 검증하기 위해 두 알고리즘을 함께 적용한 경우에 대해서도 실험 결과를 구하였다. 압축 효율 및 부호화 연산량을 비교하기 위해서는 JCT-VC에서 사용하는 공통 테스트 조건 중 가장 높은 압축 효율을 보이는 random-access configuration을 사용하였으며, 평가 영상은 [15]에 정의된 영상들 중 Class B(1920x1080) 5개, Class C(832x480) 4개, Class D(416x240) 4개 등 총 13개의 영상들을 사용하였다.

2. 보간 필터 길이 최적화

움직임 추정을 위한 보간 필터 길이를 2-tap, 4-tap, 6-tap으로 바꾸어 가면서 압축 효율 및 부호화기 복잡도를 측정한 결과는 표 4와 같다. 단, 1/2 정밀도와 1/4 정밀도에 대한 보간 필터 길이는 항상 같도록 하여 실험을 수행하였다.

표 4에서 볼 수 있듯이 필터 탭 수를 감소시킴에 따라서 압축 효율은 하락하며, 부호화기 복잡도 또한 감소하였다. 필터 탭 수가 각각 4와 6인 경우, 압축 효율은 거의 감소하지 않으면서도 부호화기 복잡도는 각각 3.2%와 7.1% 감소함을 확인할 수 있다. 필터 탭 수가 2인 경우에는 10.6%의 부호화기 복잡도를 감소시킬 수 있는 반면, 1.3% 가량 압축 효율이 감소한다. 하지만 압축 효율 감소폭은 입력 영상의 해상도가 증가할수록 감소하는 경향을 보였고, 가장 부호화기 복잡도가 문제가 되는 Class B(1920x1080) 영상에서는 필터 탭 수를 2로 한다고 해도 0.7% 가량의 압축 효율만이 감소하였다.

표 4. 다양한 보간 필터 탭 수에 대한 압축 효율 및 부호화 시간

Table 4. Coding efficiency and relative encoding time of various number of filter taps.

Filter taps	BD-Bitrate (%)				Rel. enc. time (%)
	Class B	Class C	Class D	Average	
6	0.0	0.1	0.1	0.1	96.8
4	0.1	0.2	0.3	0.2	92.9
2	0.7	1.4	2.0	1.3	89.4

3. 고속 탐색 전략

표 5는 비정수 움직임 벡터 탐색 과정에 있어서 제안한 diamond search와 band search를 사용한 경우 각각에 대한 압축 효율 및 부호화 시간을 나타낸 것이다. 단, 1/2 정밀도와 1/4정밀도에 동일한 알고리즘을 적용한 결과이다.

표 5. Diamond search와 band search에 대한 압축 효율 및 부호화 시간

Table 5. Coding efficiency and relative encoding time of diamond search and band search.

Search type	BD-Bitrate (%)				Rel. enc. time (%)
	Class B	Class C	Class D	Average	
Diamond	0.4	0.7	0.8	0.6	95.4
Band	0.9	1.4	1.6	1.3	92.9

표 5에서 diamond search를 사용하면 평균 0.6%의 압축 효율이 하락하는 반면 약 4.6%의 부호화 시간이 감소되었으며, band search를 사용하면 평균 1.3%의 압축 효율이 하락하는 반면 약 7.1%의 부호화 시간이 감소되었다.

4. 보간 필터 길이 최적화와 고속 탐색 전략 결합

표 6은 보간 필터 길이 최적화와 고속 탐색 전략을 함께 적용하는 경우에 대한 결과이다.

표 6의 결과를 보면 평균 0.7%, 1.5%, 2.5%, Class B(1920x1080) 0.4%, 1.1%, 1.6% 압축 효율 하락 수준에서 부호화기 복잡도는 각각 11.9%, 14.4%, 17.4% 감소하였다. 또한, 두 가지 제안한 방법, 즉 보간 필터 길이 최적화와 고속 탐색 전략의 효과는 서로 독립적으로 더해지는 것으로 나타났다.

표 6. 보간 필터 길이 최적화와 고속 탐색 전략을 결합한 경우에 대한 압축 효율 및 부호화 시간
Table 6. Coding efficiency and relative encoding time of filter tap optimization and fast search strategy.

Filter taps	Search type	BD-Bitrate (%)				Rel. enc. time (%)
		Class B	Class C	Class D	Average	
6	Diamond	0.4	0.7	0.8	0.6	92.2
	Band	1.0	1.5	1.6	1.4	89.4
4	Diamond	0.4	0.9	0.9	0.7	88.1
	Band	1.1	1.6	1.8	1.5	85.6
2	Diamond	1.0	1.9	2.7	1.9	85.2
	Band	1.6	2.6	3.3	2.5	82.6

5. 선택적 보간 블록 생성

제 Ⅲ장 3절에서 설명한 바와 같이 diamond search와 band search에서는 보간 블록 생성을 선택적으로 할 수 있기 때문에, 앞 절의 결과에 추가적으로 반영한 결과를 표 7에 정리하였다.

표 7은 제안한 알고리즘 모두를 적용한 결과이다. 표 6의 결과와 비교해 볼 때, 선택적 보간 블록 생성 방법은 약 1.3%에서 5.4% 정도의 부호화 복잡도를 추가로 감소시킴으로써, 평균 0.7%, 1.5%, 2.5%, Class B (1920x1080) 0.4%, 1.1%, 1.6% 압축 효율 하락 수준에서 부호화기 복잡도는 각각 13.6%, 18.5%, 21.1% 감소하였다.

표 7. 보간 필터 길이 최적화, 고속 탐색 전략, 선택적 보간 블록을 결합한 경우에 대한 압축 효율 및 부호화 시간
Table 7. Coding efficiency and relative encoding time of filter tap optimization, fast search strategy and selective interpolation.

Filter taps	Search type	BD-Bitrate (%)				Rel. enc. time (%)
		Class B	Class C	Class D	Average	
6	Diamond	0.4	0.7	0.8	0.6	89.9
	Band	1.0	1.5	1.6	1.4	84.0
4	Diamond	0.4	0.9	0.9	0.7	86.4
	Band	1.1	1.6	1.8	1.5	81.5
2	Diamond	1.0	1.9	2.7	1.9	83.9
	Band	1.6	2.6	3.3	2.5	78.9

6. 고속 최적 분할구조 탐색 알고리즘 병행 적용

HM15.0에는 고속 최적 분할 구조 탐색 알고리즘(early skip decision)이 구현되어 있다. 표 8은 이 알고리즘이 적용된 상황에서 추가적으로 제안한 방법 중 필터 탭수 4, band search, 선택적 보간 블록 생성 방법을 적용한 경우에 대한 실험 결과이다. 성능 및 부호화 시간 비교 대상은 HM15.0에 고속 최적 분할 구조 탐색 알고리즘만을 적용한 경우로 하였다.

표 8의 결과를 표 7과 비교해 보면, 압축 효율 하락이 1.5%에서 1.4%로, 부호화 시간도 81.9%에서 81.7%로 거의 유사하게 나타남으로써 제안한 방법이 고속 최적 분할 구조 탐색 알고리즘의 사용 여부와 관계없이 유사한 복잡도 감소 효과를 갖는 것을 보였다.

표 8. 제안한 방법을 고속 최적 분할 구조 탐색 알고리즘과 함께 사용한 경우의 압축 효율 및 부호화 시간
Table 8. Coding efficiency and relative encoding time of proposed methods with early skip decision.

Filter taps	Search type	BD-Bitrate (%)				Rel. enc. time (%)
		Class B	Class C	Class D	Average	
4	Band	1.0	1.5	1.7	1.4	81.7

Ⅴ. 결 론

본 논문에서는 HEVC의 비정수 움직임 벡터 예측 단계를 구성하는 요소들을 단계별로 분석하고 부호화기 복잡도를 감소시키기 위한 필터 탭 수 최적화 및 두 가

지 고속 탐색 전략들, 그리고 고속 탐색 전략에 따라 선택적으로 보간 블록을 생성하는 방법을 제안하였다. 실험 결과, 제안한 방법을 사용하면 평균 압축 성능 하락 폭 0.7%, 1.5%, 2.5%에서 부호화기 복잡도를 각각 13.6%, 18.5%, 21.1% 감소시킬 수 있었다. 또한 고해상도 영상(1920x1080)의 경우 압축 성능 하락 폭이 0.4%, 1.1%, 1.6%로 감소함으로써 제안한 방법이 고해상도 영상에 더욱 효과적임을 보였다. 한편 제안한 방법은 HM15.0에 구현되어 있는 고속 최적 분할 구조 탐색 방법과 함께 사용되는 경우에도 유사한 수준의 효과를 나타내었다.

REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 22, No. 12, pp. 1649-1668, Dec. 2012.

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 7, pp. 560-576, Aug. 2003.

[3] E. Ohwovoriole, Y. Andreopoulos, "Rate-distortion performance of contemporary video codecs: Comparison of Google/WebM VP8, AVC/H.264, and HEVC TMuC", *LENS Symp.*, London, Sep. 2010.

[4] F. De Simone, L. Goldmann, J.-S. Lee, T. Ebrahimi, "Performance analysis of VP8 image and video compression based on subjective evaluations," *SPIE Appl. Digital Image Proc. XXXIV*, Aug. 2011.

[5] Woo-Jin Han, Junghye Min, Il-Koo Kim, Elena Alshina, Alexander Alshin, Tammy Lee, Jianle Chen, Vadim Seregin, Sunil Lee, Yoon-Mi Hong, Min-Su Cheon, Nikolay Shlyakhov, Ken McCann, Thomas Davies and Jeong-Hoon Park, "Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 20, No. 12, pp. 1709-1720, Dec. 2010.

[6] R. H. Gweon, Y. L. Lee, J. Lim, "Early termination of CU encoding to reduce HEVC complexity," JCTVC-F045, 6th JCT-VC meeting, Jul. 2011, Torino, Italy.

[7] J. Kim, G. Kim and C. Yim, "CU size decision method based on statistics for HEVC encoding," Proc. of the Institute of Electronics and Information Engineers of Korea, pp. 533-535, Nov. 2013.

[8] X. Shen, L. Yu and J. Chen, "Fast coding unit size selection for HEVC based on Bayesian decision rule," *Proc. of Picture Coding Symposium (PCS)*, pp. 453-456, May 2012.

[9] S. Yoo, Y. Ahn and D. Sim, "Fast HEVC encoding based on CU-depth first decision," Journal of the Institute of Electronics and Information Engineers of Korea, Vol. 49, SP No. 3, pp. 40-50, 2012.

[10] W. Jiang, H. Ma, Y. Chen, "Gradient based fast mode decision algorithm for intra prediction in HEVC," *Proc. of Consumer Electronics, Communications and Networks (CECNet)*, pp. 1836-1840, Apr. 2012.

[11] L. Zhao, L. Zhang, S. Ma and D. Zhao, "Fast mode decision algorithm for intra prediction in HEVC," *Proc. of Visual Communications and Image Processing (VCIP)*, pp. 1-4, Nov. 2011.

[12] Z. Chen, C. Du, J. Wang, Y. He, "PPFPS-a paraboloid prediction based fractional pel search strategy for H.26L," *Proc. of IEEE ISCAS*, Vol. 3, III-9-III-12, USA, 2002.

[13] Z. Chen, J. Xu, Y. He, J. Zheng, "Fast integer-pel and fractional-pel motion estimation for H.264/AVC," *Journal of Visual Communication and Image Representation*, Vol. 17, Issue 2, pp. 264-290, Apr 2006.

[14] Kemal Ugur, Alexander Alshin, Elena Alshina, Frank Bossen, Woo-Jin Han, Jeong-Hoon Park and Jani Lainema, "Motion Compensated Prediction and Interpolation Filter Design in H.265/HEVC," *IEEE Journal of Selected Topics in Signal Processing*, Vol. 7, No. 6, pp. 946-956, December 2013.

[15] HEVC reference software version 15.0 (HM15.0), https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags/HM-15.0, Jul. 2014.

[16] F. Bossen, "Common HM tst conditions and software reference configurations," in Proc. of JCTVC-L1100, 12th JCT-VC meeting, Geneva, Switzerland, Jan. 2013.

[17] G. Bjontegaard, "Calculation of average PSNR differences between RD curves," VCEG-M33, 13th VCEG meeting, Austin, TX, USA, Apr.

2001.
[18] S. Zhu and K. MA, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Transactions on Image Processing*, Vol. 9, No. 2, pp. 287-290, Feb. 2000.

저 자 소 개

한 우 진(정회원)
1995년 KAIST 전산학과 학사 졸업.
1997년 KAIST 전산학과 석사 졸업.
2002년 KAIST 전산학과 박사 졸업.
2002년 SL2 연구소장
2003년~2010년 삼성전자 DMC연구소 수석연구원
2011년~현재 가천대학교 소프트웨어학과 조교수
<주관심분야 : 영상압축, 영상이해, 신호처리>