

정규논문 (Regular Paper)
방송공학회논문지 제20권 제1호, 2015년 1월 (JBE Vol. 20, No. 1, January 2015)
<http://dx.doi.org/10.5909/JBE.2015.20.1.171>
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

HEVC 부호화기에서의 화면 간 예측모드 고속 결정

한우진^{a)†}, 안준형^{a)}, 이종호^{a)}

Early Decision of Inter-prediction Modes in HEVC Encoder

Woo-Jin Han^{a)†}, Joon-Hyung Ahn^{a)} and Jong-Ho Lee^{a)}

요약

HEVC는 H.264/AVC에 비해 압축 성능을 크게 개선시킬 수 있지만 부호화기와 복호화기 모두 복잡도가 크게 증가한다. 본 논문에서는 HEVC의 화면 간 예측 모드 결정 과정을 분석하고, 이 결과로부터 부호화기 및 복호화기의 복잡도를 효과적으로 감소시키기 위한 방법을 제안하였다. 제안하는 방법은 단방향 예측 모드의 결과로부터 양방향 예측 모드를 수행하지 않아도 되는 조건을 찾고, 이 조건을 만족하는 경우 미리 종료시킴으로써 부호화 복잡도를 감소시킨다. 실험 결과 압축률 하락 폭이 각각 0.6%, 1.0%, 1.5%인 경우 부호화 복잡도를 12.0%, 14.2%, 17.2% 감소시킬 수 있었으며, 이 때, 양방향 예측 모드의 비율을 각각 6.3%, 11.8%, 16.6% 감소시킴으로써 복호화기의 복잡도도 함께 감소시킬 수 있었다. 마지막으로, 제안한 방법이 HEVC 참조 소프트웨어에 기 적용되어 있는 고속화 알고리즘과 함께 사용되는 경우에도 유사한 효과를 낼 수 있음을 검증하였다.

Abstract

HEVC can increase the coding efficiency significantly compared with H.264/AVC however it requires much larger computational complexities in both encoder and decoder. In this paper, the decision process of inter-prediction modes in the HEVC reference software has been studied and a fast algorithm to reduce the computational complexity of encoder and decoder is introduced. The proposed scheme introduces a early decision criteria using the outputs of uni-directional predictions to skip the bi-directional prediction estimation. From the experimental results, it was proven that the proposed method can reduce the encoding complexity by 12.0%, 14.6% and 17.2% with 0.6%, 1.0% and 1.5% of coding efficiency penalty, respectively. In addition, the ratio of bi-directional prediction mode was reduced by 6.3%, 11.8% and 16.6% at the same level of coding efficiency penalty, respectively, which should lead to the decoder complexity reduction. Finally, the effects of the proposed scheme are maintained regardless of the use of the early skip decision algorithm which is implemented in the HEVC reference software.

Keyword: HEVC, encoder, inter-prediction, fast algorithm

1. 서론

최근 표준화가 완료된 HEVC^[1]는 H.264/AVC^[2] 대비 압축 효율을 약 40% 이상 향상시킬 수 있는 것으로 알려져 있지만^{[3][4][5]} 부호화 및 복호화 과정의 복잡도가 크게 증가하기 때문에 이를 감소시키기 위한 연구들이 활발하게 제

a) 가천대학교 IT대학 소프트웨어학과(Dept. of Software, College of IT, Gachon University)

† Corresponding Author : 한우진(Woo-Jin Han)
E-mail: hurumi@gmail.com
Tel: +82-31-750-8668
ORCID: <http://orcid.org/0000-0001-5114-4117>

Manuscript received November 12, 2014 Revised December 2, 2014
Accepted December 2, 2014

안되어 왔다.

하나의 slice를 16x16크기의 매크로블록(macroblock; MB)들로 분할하여 처리하는 H.264/AVC에 비해, HEVC에서는 최대 64x64까지의 크기를 가질 수 있는 coding tree unit(CTU)들로 slice를 분할하고, 각각의 CTU들이 쿼드트리 형태로 다시 재귀적 분할되는 자유로운 형태를 갖고 있다. 이렇게 분할된 단위를 coding unit(CU)이라고 하며, 이 CU들은 추가적으로 prediction unit(PU)으로 분할된다^[6]. 이러한 자유로운 분할 형태 때문에 HEVC에서는 최적 분할 구조를 찾기 위해 많은 연산량이 요구되며, 이를 감소시키기 위한 많은 연구들이 발표되었다. [7][8]에서는 잔차신호의 모든 변환 계수가 양자화 과정에서 0이 되는 경우 더 이상의 재귀적 분할을 하지 않도록 제한함으로써 부호화 복잡도를 크게 낮추었다. 이 기술은 HEVC의 참조 소프트웨어에 early skip decision(ESD)라는 이름으로 채택되어 있다. [9]에서는 주변에 위치한 블록들의 분할 정보를 활용하여 현재 블록의 분할 가능한 범위를 제한하였으며, [10]에서는 Bayesian decision rule을 이용하여 최적 CU의 크기를 결정하는 방법을 적용하였다.

HEVC 부호화를 고속화함에 있어 최적 재귀적 분할 구조를 고속으로 추정하기 위한 방법들이 주로 제안되었지만, 이외에도 특정 coding tool의 부호화 과정을 단순화시키기 위한 연구들도 발표되었다. [11]에서는 HEVC의 화면 내 예측을 수행함에 있어 gradient 기반의 고속화 방법을 제안하였고, [12]에서는 주변 intra 블록의 direction 정보를 활용하여 화면 내 예측의 부호화 과정 복잡도를 낮춤으로써 모든 픽처가 intra-prediction으로 부호화되는 경우 상당한 수준의 복잡도 감소를 달성하였다.

일반적으로 부호화 단계에서 화면 간 예측이 차지하는 연산량이 화면 내 예측에 비해서 월등히 높고, 압축 성능에 미치는 영향 또한 훨씬 크기 때문에 화면 간 예측의 복잡도를 낮추는 것은 매우 중요하다. [13]에서는 다양한 분할 크기에 대해 움직임 추정을 수행할 때, edge intensity와 temporal stationarity를 이용하여 분할 크기의 후보 수를 줄이는 방법으로 복잡도를 낮추었으며, [14]에서는 motion activity를 이용하여 추정해야 하는 총 모드의 수를 줄임으로써 복잡도를 낮추었다. 그러나 이러한 연구들 대부분은

H.264/AVC를 그 대상으로 하고 있으며, 최적 분할 크기를 찾기 위한 탐색 과정을 단순화 시키는 것을 목표로 하기 때문에 HEVC에서 이미 제안된 최적 분할 방법^{[7][8][9][10]}들과 그 유효성이 중복된다.

본 논문에서는 HEVC 부호화기의 화면 간 예측 모드 결정 과정을 분석하고, 이를 구성하는 요소들의 성능 및 복잡도를 분석하였다. 또한, 두 개의 참조 픽처로부터 얻은 블록들의 가중 합을 계산하여 현재 픽처의 블록을 예측하는 기술인 양방향 예측 모드를 적응적으로 생략할 수 있는 방법을 고안함으로써 복호화 및 부호화 복잡도를 낮추기 위한 연구를 수행하였다. 한편, 독립적으로 사용되었을 경우는 물론 HEVC 참조 소프트웨어에 적용되어 있는 ESD 알고리즘과 함께 적용하였을 경우에 대해서도 그 결과를 보인다.

제 II장에서는 HEVC의 화면 간 예측 부호화 과정을 개략적으로 소개하며, 제 III장에서는 단방향/양방향 예측 모드의 성능과 복잡도를 분석하였다. 제 IV장에서는 양방향 예측 모드의 부호화 복잡도를 감소시키기 위한 조건을 제안하였으며, 제 V장에서 이에 대한 실험 결과를 보인 후, 마지막으로 제 VI장에서 결론을 맺는다.

II. HEVC 부호화기에서의 화면 간 예측 부호화 과정

HEVC의 부호화기에서는 slice를 최소 8x8에서 최대 64x64까지의 크기를 가질 수 있는 CTU들로 분할한 뒤, 각 CTU들을 쿼드트리 형태로 다시 재귀적 분할한다. 이 때 분할된 각각의 영역인 CU들에 대해, 화면 내 예측 부호화 혹은 화면 간 예측 부호화 중 최적의 방법을 선택하게 된다. 화면 간 예측 부호화 과정은 CU를 다시 다양한 크기로 분할한 단위인 PU를 기준으로 수행되는데, HEVC는 2Nx2N, Nx2N, 2NxN, NxN, N/2x2N(L), N/2x2N(R), 2NxN/2(U), 2NxN/2(D)의 총 8가지 PU 분할 모양을 갖고 있다^[1]. 화면 간 예측 부호화 과정에서는 주어진 PU 크기에 대해 화면 간 예측 부호화에 필요한 파라미터들, 즉 최적 움직임 벡터, 참조 픽처 번호, 단방향/양방향 예측 모드를 추정한다.

HEVC에서는 advanced motion vector prediction (AMVP) 모드와 merge 모드의 두 가지 서로 다른 파라미터 부호화 방법을 갖고 있다. AMVP 모드의 경우, 복수 개의 움직임 벡터 후보 중 최적 값을 가리키는 인덱스, 움직임 벡터 차분 값, 참조 픽처 번호, 단방향/양방향 예측 모드를 부호화하는데 비해, merge 모드의 경우 복수 개의 움직임 벡터 후보 중 최적 값을 가리키는 인덱스만 부호화하며, 다른 값들은 모두 예측된 값들을 그대로 사용한다. AMVP 모드는 파라미터들을 자유롭게 결정하여 부호화할 수 있는 장점이 있는 반면 파라미터 부호화에 필요한 비트수가 많고, 복잡한 부호화 과정인 움직임 추정 과정이 필요하다. Merge 모드는 부호화에 필요한 비트수가 매우 적으며 움직임 추정 과정이 필요하지 않지만 예측 값이 부정확할 경우에는 사용될 수 없는 단점이 있다. 다음 절에서는 각 모드의 부호화 과정을 상세히 설명한다.

1. AMVP 모드 부호화 과정

AMVP 모드에서는 부호화하고자 하는 현재 블록의 주위에 위치한 블록 및 시간적으로 인접한 블록으로부터 움직임 벡터 후보들을 선정한다. 그 후, 움직임 추정 단계를 수행하여 최적의 움직임 벡터, 참조 픽처 번호, 단방향/양방향 예측 모드를 구한다. AMVP 모드의 부호화 단계에서는 L0 리스트를 이용하는 단방향 탐색, L1 리스트를 이용하는 단방향 탐색, L0 리스트와 L1 리스트를 모두 이용하는 양방향 탐색 순으로 움직임 추정 단계가 진행된다.

양방향 모드 움직임 추정 단계에서는 복잡도를 낮추기 위해서 L0 리스트의 참조 픽처와 L1 리스트의 참조 픽처 간 모든 조합에 대한 움직임 추정을 수행하는 대신, 하나를 고정하고 나머지 하나를 최적화하는 과정을 반복적으로 수행한다. 단, HEVC 참조 소프트웨어에서는 이러한 반복 과

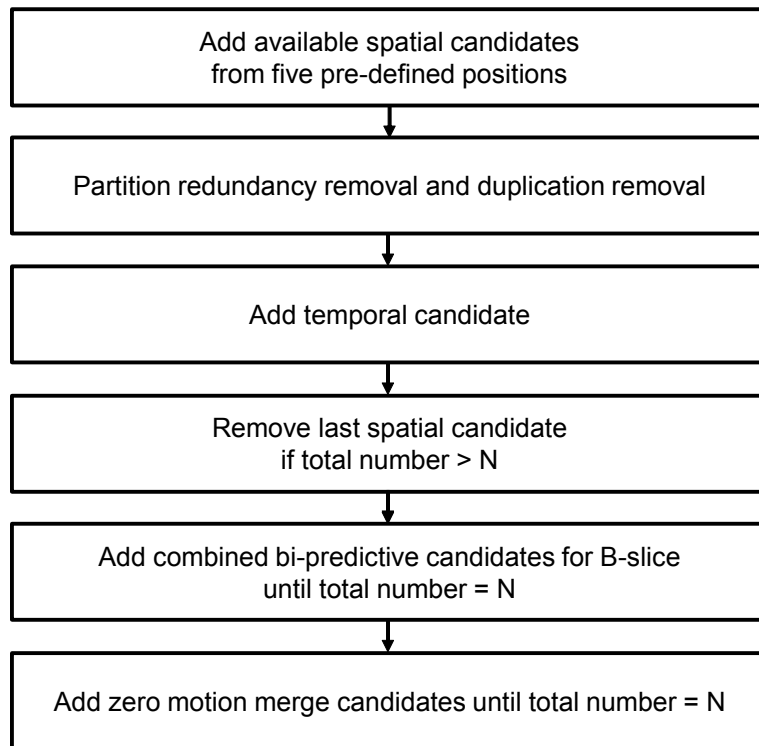


그림 1. AMVP 모드의 부호화 과정
Fig. 1. Encoding process of AMVP mode

정에서의 복잡도를 낮추기 위해서 반복 회수를 1번으로 제한하고 있으며, L0 리스트의 참조 픽처와 L1리스트의 참조 픽처가 동일한 경우 L1 리스트에 대한 움직임 추정을 생략한다. <그림 1>은 AMVP의 부호화 과정을 그림으로 나타낸 것이다.

2. Merge 모드 부호화 과정

Merge 모드에서는 부호화하고자 하는 현재 블록과 공간적으로 인접한 블록들 및 시간적으로 인접한 블록으로부터 움직임 벡터, 참조 픽처 번호, 단방향/양방향 예측 모드를 얻어 복수 개의 merge candidate들을 구성한다. 움직임 벡터 정보만을 예측한 후, 최적 움직임 벡터를 별도로 추정하는 과정이 필요한 AMVP 모드와는 달리 merge 모드에서는 움직임 벡터, 참조 픽처 번호, 단방향/양방향 예측 모드를

최적 merge candidate로부터 가져와서 그대로 사용하기 때문에, 추가적인 움직임 추정 단계가 필요하지 않다.

사용되는 merge candidate들의 총 개수는 slice 별로 1~5 사이의 값 중 하나로 정할 수 있다. 이 값을 N이라고 할 때, 공간적으로 인접한 블록들로부터 확보된 merge candidate의 수가 N보다 작은 경우 먼저 참조 픽처 중 시간적으로 인접한 블록의 움직임 정보를 merge candidate로 추가하고, 그 후 미리 정해진 규칙에 따라서 merge candidate들을 생성하여 추가한다. 먼저, 이미 존재하는 L0 단방향 merge candidate와 L1 단방향 merge candidate들을 미리 정해진 순서로 조합하여 새로운 양방향 merge candidate들을 만들어낸다. 이러한 과정을 combined bi-predictive merge candidate라고 하며, 이 과정 후에도 총 확보된 merge candidate의 수가 N보다 작은 경우 zero merge candidate, 즉 움직임 벡터 (0, 0) 및 첫 번째 참조 픽처를 사용하는 merge

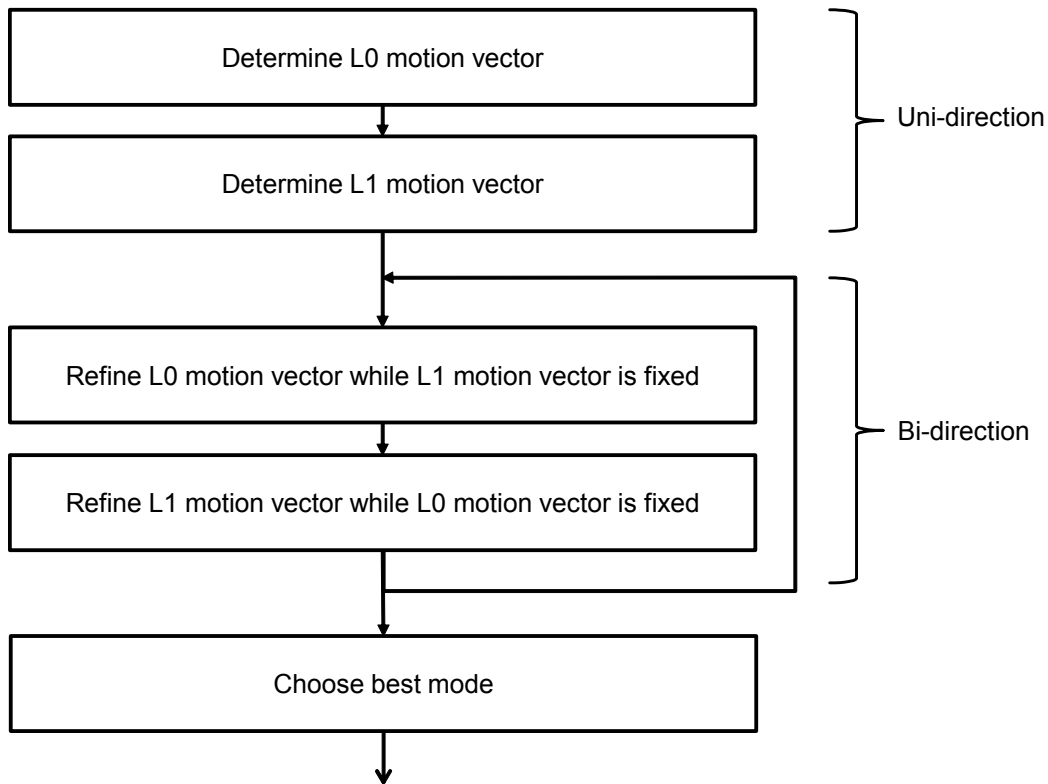


그림 2. merge candidate list 구성 과정
 Fig. 2. construction process of merge candidate list

candidate를 추가하여 총 merge candidate의 수를 N으로 고정시킨다. 이 때, P-slice의 경우 단방향 모드로, B-slice의 경우 양방향 모드로 결정한다. <그림 2>는 merge candidate를 구성하는 과정을 보인 것이다.

Merge candidate N개가 모두 구성되고 나면, 모든 merge candidate들에 대해서 움직임 보상 과정을 수행하고, 원본 블록과의 rate-distortion cost를 계산하여 최적의 merge candidate index를 얻는다.

III. 단방향/양방향 예측 모드 성능 및 복잡도 분석

HEVC 참조 소프트웨어의 성능을 평가하기 위해서는 일반적으로 표준화 기구인 JCT-VC에서 사용하는 common test condition^[15]을 사용한다. 이 조건은 부호화 지연시간 허용 여부 및 양방향 예측 모드 사용 유무에 따라서 random-access configuration, low-delay configuration, low-delay P configuration의 세 가지로 나누어지는데, 이 중 저지연 부호화에 의한 성능 저하가 없어 가장 높은 성능을 보이는 random-access configuration을 사용하여 화면 간 예측 모드를 구성하는 요소들 중 단방향/양방향 예측 모드에 관한 성능 및 연산량 분석을 하고자 한다.

표 1. 테스트 영상
 Table 1. Test video sequences

Class	Name	Resolution	Frame-rates
B (1920x1080)	Kimono	1920x1080	24
	ParkScene		24
	Cactus		50
	BasketballDrive		50
	BQTerrace		60
C (832x480)	BasketballDrill	832x480	50
	BQMall		60
	PartyScene		50
	RaceHorses		30
D (416x240)	BasketballPass	416x240	50
	BQSquare		60
	BlowingBubbles		50
	RaceHorses		30

1. AMVP의 양방향 예측 모드

테스트를 위해 <표 1>과 같이 JCT-VC에서 사용하는 영상들 중 Class B, Class C, Class D에 해당하는 총 13개 영상을 사용하였으며 HEVC 참조 소프트웨어로는 HM15.0^[16]을 사용하였다.

먼저, 부호화기를 수정하여 AMVP 모드에서 양방향 예측 모드를 수행하지 않도록 변경한 후 그 성능을 확인해 보았다. 압축 성능은 HM15.0에 대한 BD-rates^[17] 값을 사용하였으며, 복잡도는 HM15.0 대비 상대적 부호화 시간을 백분율로 표시하였다. <표 2>는 이 결과를 요약한 것이며, 기존 HM15.0의 성능을 기준으로 표시하였으므로 BD-rates 상의 양수 값은 성능이 하락했음을 의미하며, Encoding time이 100보다 작으면 복잡도가 감소했음을 의미한다.

표 2. AMVP에서 양방향 예측 모드를 사용하지 않는 경우 (AMVP-NO-BI로 표시) 압축 성능 및 복잡도
 Table 2. Coding efficiency and encoding time of AMVP mode without bi-directional prediction

	Class B	Class C	Class D	Average	Encoding Time (%)
AMVP-NO-BI	5.7	3.6	4.3	4.5	90.7

<표 2>의 결과로부터 AMVP의 양방향 예측 모드는 약 4.5% 정도 성능 향상에 기여하며 총 부호화 시간 중 약 9.3% 정도를 차지함을 알 수 있다. 한편, 이렇게 AMVP의 양방향 예측 모드를 제거한 경우, 전체 화면 간 예측 모드 중 양방향 예측 모드가 차지하는 비율이 얼마나 감소하는지 알기 위해서 압축된 스트림을 분석하여 양방향 예측 모드의 비율을 측정해 보았다. <표 3>은 그 결과이다.

표 3. AMVP-NO-BI의 양방향 예측 모드 비율
 Table 3. Ratio of bi-directional prediction of AMVP-NO-BI

	Class B	Class C	Class D	Average
HM15.0	60.8	45.8	51.6	53.3
AMVP-NO-BI	55.8	38.8	44.8	47.2
Difference	-5.0	-7.0	-6.8	-6.1

<표 3>의 결과로부터 AMVP에서 양방향 예측 모드를 완전히 제거하더라도 평균적으로 볼 때 6.1%의 양방향 예측 모드 비율이 감소할 뿐이며 여전히 양방향 예측 모드는 전체 inter-prediction 예측 모드 중 47.2% 가까이 사용되고 있음을 알 수 있다. 이는 AMVP 모드가 아니라 merge 모드에서 여전히 양방향 예측 모드가 많이 사용되고 있다는 것을 의미한다. 이를 확인하기 위해서는 merge 모드에서도 양방향 예측 모드를 제거한 후 실험 데이터를 구해야 하나, merge 모드의 경우 단방향/양방향 예측 모드를 별도로 부호화하는 대신 merge candidate의 단방향/양방향 예측 모드를 그대로 사용하기 때문에 HEVC 표준에 대한 수정 없이는 양방향 예측 모드만을 제거할 수가 없다. 따라서 다른 방법을 사용하여 간접적으로 분석을 수행하였다.

Merge 모드의 candidate 순서를 살펴보면 먼저 공간적으로 주변에 위치한 블록들을 우선적으로 추가하며, 그 후 시간적으로 인접한 블록, 그리고 combined bi-predictive candidate와 zero motion merge candidate를 추가한다. 특히, B-slice에서는 양방향 예측 모드인 combined bi-predictive candidate가 사용되며, zero motion merge candidate 또한 양방향 예측 모드로 설정되기 때문에 공간적/시간적으로 인접한 merge candidate들이 모두 단방향 예측 모드를 사용한다고 하더라도 양방향 예측 모드를 완전히 제거할 수 없다. Combined bi-predictive candidate와 zero merge candidate는 merge candidate list에서 후반부에 위치하므로, 이러

표 4. AMVP-NO-BI에서 총 merge candidate의 수(MC로 표시)를 변경해가면서 구한 양방향 예측 모드 비율 및 압축 성능
Table 4. Ratio of bi-directional prediction and coding efficiency of AMVP-NO-BI with respect to various number of merge candidates

	Class B	Class C	Class D	Average	BD-rates (%)
HM15.0	60.8	45.8	51.6	53.3	-
AMVP-NO-BI (MC=5)	55.8	38.8	44.8	47.2	4.5
AMVP-NO-BI (MC=4)	54.0	36.7	42.4	45.1	5.0
AMVP-NO-BI (MC=3)	49.9	31.7	38.0	40.7	6.2
AMVP-NO-BI (MC=2)	38.1	23.2	30.1	31.0	8.9
AMVP-NO-BI (MC=1)	8.9	5.1	12.1	8.7	16.5

한 효과를 검증하기 위해서 <표 4>와 같이 총 merge candidate의 수를 감소시켜가면서 그 영향을 분석해 보았다.

<표 4>의 결과에서 볼 수 있듯이 총 merge candidate의 수가 감소함에 따라서 전체 화면 간 예측 모드 중 양방향 예측 모드가 차지하는 비율이 감소하며, 압축 성능 하락도 커지는 것을 확인할 수 있다. 양방향 예측 모드는 단방향 예측 모드에 비해 복호화에 많은 복잡도를 요구하기 때문에 양방향 예측 모드 비율 감소는 복호화가 복잡도 감소와 관련이 있다. 다만 이에 따른 압축 성능 하락이 매우 크게 나타나, 압축 성능 하락을 최소화 하면서도 양방향 예측 모드 비율을 감소시킬 수 있는 방법이 필요하다.

2. AMVP의 L1 단방향 예측 모드

AMVP의 양방향 예측 모드를 제거한 AMVP-NO-BI에서 L1 단방향 예측 모드를 함께 제거하면 B-slice 대신 P-slice를 사용한 것과 유사한 형태가 된다. <표 5>에서는 이러한 조건 하에서의 성능 및 복잡도를 측정해 보았다. 비교를 위해 AMVP-NO-BI의 결과 및 두 결과의 차이를 함께 표시하였다.

표 5. AMVP에서 양방향 예측 모드/L1 단방향 예측 모드를 사용하지 않는 경우 (AMVP-NO-BI-L1으로 표시)의 성능/복잡도
Table 5. Coding efficiency and encoding time of AMVP without both bi-directional prediction and L1 uni-directional prediction

	Class B	Class C	Class D	Average	Encoding Time (%)
AMVP-NO-BI	5.7	3.6	4.3	4.5	90.7
AMVP-NO-BI-L1	11.6	15.2	13.1	13.3	80.0
Difference	5.9	11.6	8.8	8.8	10.7

<표 5>에서 볼 수 있듯이 AMVP의 L1 단방향 예측 모드와 양방향 예측 모드를 모두 사용하지 않는 경우 평균 약 13.3%의 성능 하락이 관측되었으며 부호화 시간은 20.0% 가량 감소하였다. 다만 AMVP의 양방향 예측 모드 자체가 L0 단방향 예측을 사용한 예측 블록과 L1 단방향 예측을 사용한 예측 블록을 필요로 하므로, 양방향 예측 모드를 유

지하면서 L1 단방향 예측 모드만을 제거한 결과를 얻을 수 없다. 대신 <표 5>의 AMVP-NO-BI와 AMVP-NO-BI-L1 간 차이로부터 AMVP의 L1 단방향 예측 모드만의 효과를 추정할 수 있는데, 결과적으로 AMVP의 L1 단방향 예측 모드는 약 8.8%의 성능 하락에 기여함으로써 4.5%의 성능 하락에 기여한 AMVP의 양방향 예측 모드에 비해 약 4.3% 가량 더 큰 영향을 주었음을 알 수 있다. 한편, 복잡도 측면에서 볼 때, AMVP의 L1 단방향 예측 모드는 전체 부호화 시간 기준 약 10.7%, 양방향 예측 모드는 약 9.3%에 해당하므로, 약 1.4% 정도 AMVP의 단방향 예측 모드가 더 많이 차지한다고 볼 수 있다.

이러한 관측 결과로부터, AMVP의 양방향 예측 모드는 L1 단방향 예측 모드와 비교할 때 성능 향상 폭은 반 정도이지만 부호화 복잡도는 유사하다고 판단할 수 있다. 즉, 동일 성능 향상 폭 대비 부호화 복잡도가 높다고 할 수 있으므로, 본 논문에서는 반드시 필요한 경우에만 양방향 예측 모드의 부호화 과정을 수행함으로써 부호화 복잡도를 낮추고자 한다. 제 IV장에서는 AMVP 양방향 예측 모드의 복잡도를 낮추기 위한 최적화 방법을 제안한다.

IV. AMVP 양방향 예측 모드 고속화

1. AMVP 양방향 예측 모드 수행 조건 예측

제 2장에서 보인 AMVP 부호화 과정은 L0 단방향 예측, L1 단방향 예측, 그리고 양방향 예측 순으로 진행된다. 본 논문에서는 L0 단방향 예측과 L1 단방향 예측이 끝난 후 그 결과를 이용하여 양방향 예측을 수행하지 않아도 되는 조건을 찾음으로써 AMVP의 양방향 예측 부호화 과정에 소요되는 연산량을 감소시키고자 한다.

HEVC 참조 소프트웨어에 이미 구현되어 있는 early skip decision(ESD)^{[7][8]}에서는 2Nx2N 화면 간 예측 부호화를 수행한 후 coded block flag(CBF)의 값을 검사하여 0인 경우 이후의 화면 간 예측 부호화 과정을 생략하도록 되어 있다. 다만, CBF를 검사하기 위해서는 잔차 신호에 대한 변환 및 양자화 과정을 거쳐야 하는데, AMVP 부호화 과정에서의

L0 단방향 예측, L1 단방향 예측은 복잡도를 낮추기 위해 잔차 신호 변환 및 양자화를 생략한 후 근사된 RD-cost를 이용하여 수행되므로 CBF 검사를 할 수가 없고, 변환 및 양자화 과정을 거치도록 수정한다면 추가적인 연산량이 필요한 문제가 있다. 이를 해결하기 위해서는 L0 단방향 예측과 L1 단방향 예측을 하기 위해서 사용되는 중간 정보들을 최대한 활용하고, 단순한 계산만으로 양방향 예측 모드의 수행이 필요한지의 여부를 판단하여야 한다.

크기 NxM을 갖는 PU에 대해 L0 단방향 예측을 수행하여 얻은 최적 예측 블록을 P0(x, y), L1 단방향 예측을 수행하여 얻은 최적 예측 블록을 P1(x, y)라고 하자. 이 두 블록들은 각각 L0 단방향 예측과 L1 단방향 예측을 수행하는 과정에서 이미 얻어지기 때문에 움직임 보상과 같은 추가 연산이 필요하지 않다. 한편, 두 블록을 평균하여 얻은 블록 A(x, y)는 식 (1)과 같이 계산된다. Weighted prediction은 적용되지 않는다고 가정하였으나, 이에 대한 확장은 쉽게 가능하다.

$$A(x, y) = (P_0(x, y) + P_1(x, y) + 1) \gg 1 \quad (1)$$

A(x, y)는 단방향 예측에 최적화된 움직임 벡터들로 만든 양방향 예측 블록이며, HEVC의 부호화기에서는 이 움직임 벡터들의 값을 재조정해가면서 최적 값을 찾기 때문에 A(x, y)는 최종적으로 결정되는 최적 양방향 예측 블록의 초기 근사치라고 생각할 수 있다. 원본 블록을 O(x, y), AMVP의 양방향 예측이 완료된 후 얻은 최적 양방향 예측 블록을 B(x, y)라고 하면, P0(x, y), P1(x, y), A(x, y), B(x, y) 각각에 대한 원본 블록과의 RD-cost RD0, RD1, RDA, RDB는 식 (2)와 같이 근사시킬 수 있다.

$$\begin{aligned} RD_0 &= \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} |O(x, y) - P_0(x, y)| + \lambda B_0 \\ RD_1 &= \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} |O(x, y) - P_1(x, y)| + \lambda B_1 \\ RD_A &= \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} |O(x, y) - A(x, y)| + \lambda (B_0 + B_1) \\ RD_B &= \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} |O(x, y) - B(x, y)| + \lambda B_2 \end{aligned} \quad (2)$$

식 (2)에서 λ 는 Lagrangian multiplier, B0, B1은 각각 L0 단방향 움직임 벡터와 L1 단방향 움직임 벡터를 부호화하기 위해서 필요한 비트수이며, B2는 양방향 움직임 벡터의 추정이 완료된 후, 이를 부호화하기 위해 필요한 비트수이다. 또한, RDA는 단방향 예측에 최적화된 움직임 벡터들로 만든 양방향 예측 블록을 사용하므로, 필요한 비트수는 단순히 두 단방향 움직임 벡터를 부호화하기 위한 비트수를 더하여 근사하였다.

한편, B(x, y)는 A(x, y)에 비해 추가적인 움직임 벡터 최적화를 수행하여 얻은 블록이므로 A(x, y)에 비해 원본 블록과의 RD-cost가 더 작다고 가정할 수 있으며, 이 경우 식 (3)이 성립한다.

$$RD_A > RD_B \tag{3}$$

이때, 만일 RDA가 이미 단방향 최적 예측 블록을 이용하여 얻은 두 RD-cost 값들인 RD0과 RD1보다 작은 경우라면 식 (4)가 성립하며, 이는 두 단방향 최적 예측 블록들에 비해 최적 양방향 예측 블록이 더 원본 블록에 가깝다는 것, 즉 양방향 예측이 유효할 가능성이 높음을 의미한다.

$$RD_0 > RD_A > RD_B \text{ and } RD_1 > RD_A > RD_B \tag{4}$$

이로부터, 식 (5)가 만족되는 경우에 한해서만 양방향 예측을 수행하는 방법을 제안한다.

$$RD_0 > \alpha RD_A \text{ and } RD_1 > \alpha RD_A \tag{5}$$

이 때, RD0과 RD1은 L0 단방향 예측과 L1 단방향 예측 과정에서 최적 값을 탐색할 때 이미 계산되는 값이므로 별도의 계산을 요구하지 않는다. 따라서 식 (5)는 RDA의 계산만을 요구하며, 적은 연산량만으로 계산될 수 있다. 식 (5)에서 추가된 상수 α 는 양방향 예측에 얼마나 가중치를 줄 것인가를 결정하는 상수로서, $\alpha > 1$ 인 경우 RD0과 RD1에 비해 RDA의 값이 확실히 더 작은 경우(양방향 예측이 단방향 예측보다 확실히 더 좋은 경우)에 한해서만 양방향 예측을 수행한다. 따라서 α 의 값이 커질수록 양방향 예측을 더 보수적으로 수행하게 되며, 결과적으로 부호화 복잡도

가 감소하게 된다. 반대로 α 의 값이 작아질수록 양방향 예측을 더 자주 수행하게 되므로 고속화에 따른 압축 성능 하락을 최소화 할 수 있게 된다. 따라서 α 의 값을 조정함으로써 압축 성능과 복잡도 간 우선 순위를 결정할 수 있다.

2. Generalized P&B(GPB) 고속화에 관한 예외 처리

HEVC 참조 소프트웨어에는 GPB(generalized P&B) 고속화 알고리즘이 구현되어 있다. 이 알고리즘에서는 L0와 L1의 참조 픽처가 동일한 경우 L1의 움직임 벡터를 추정하는 대신, 이미 추정된 L0의 움직임 벡터와 동일하게 설정하여 복잡도를 감소시키며, 그 후 양방향 예측 모드를 통해 미세하게 조정함으로써 성능 하락을 막는다. <표 6>은 이 고속화 알고리즘을 적용하지 않은 경우의 결과이다.

표 6. GPB 고속화를 적용하지 않은 경우의 성능 및 복잡도
Table 6. Coding efficiency and encoding time of fast GPB encoding

	Class B	Class C	Class D	Average	Encoding Time (%)
NO-GPB-FAST	0.0	0.0	0.1	0.0	113.3

<표 6>에서 볼 수 있듯이, GPB 고속화는 성능 하락을 거의 가져오지 않으면서도 약 13%의 복잡도 감소 효과가 있어 매우 효과적임을 알 수 있다. 다만 이 경우, L0와 L1의 움직임 벡터 및 참조 픽처가 동일하기 때문에 식 (5)에서 RD0, RD1, RDA의 SAD부분은 항상 같고, RDA의 비트수는 항상 RD0, RD1보다 크다. 때문에 만일 고속화를 위해 $\alpha > 1$ 로 설정하면, 식 (5)는 절대 만족되지 않게 되고 이는 결국 AMVP의 양방향 예측 모드를 전혀 사용하지 않도록 하는 결과를 가져온다. 이를 해결하기 위해서, 식 (5)의 조건에서 L0와 L1의 참조 픽처가 동일한 경우 AMVP 양방향 예측 모드를 무조건 수행하도록 식 (6)과 같이 수정한다. 식 (6)에서 RefPOC0, RefPOC1은 각각 L0 참조 픽처와 L1 참조 픽처의 picture order count(POC)들을 의미한다.

$$(RD_0 > \alpha RD_A \text{ and } RD_1 > \alpha RD_A) \text{ or } (RefPOC_0 = RefPOC_1) \tag{6}$$

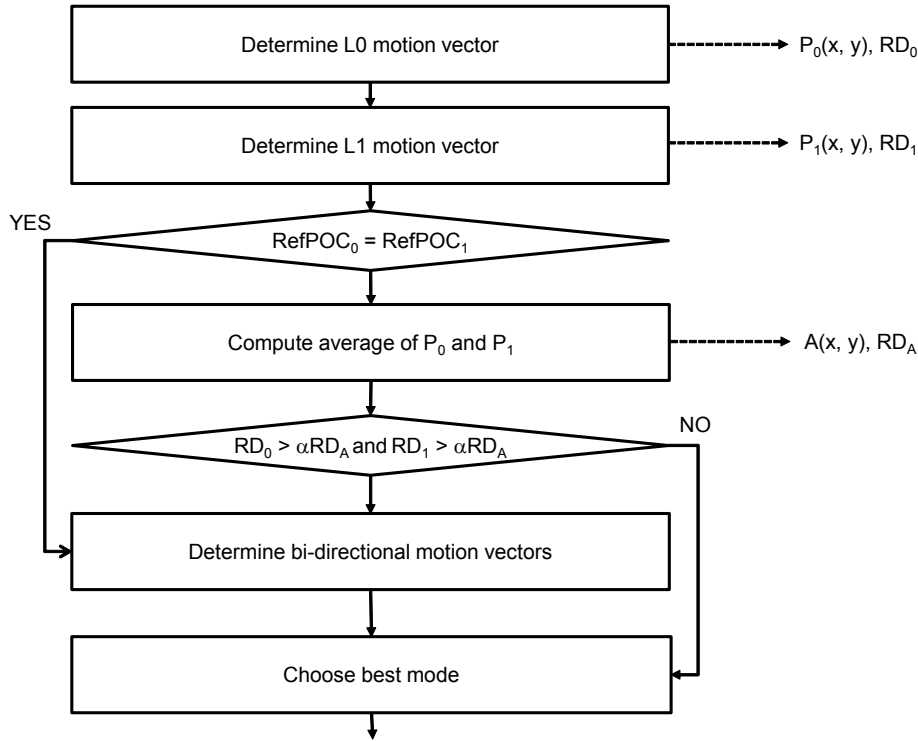


그림 3. 제안하는 AMVP 양방향 예측 고속화 알고리즘
 Fig 3. Proposed fast encoding algorithm of AMVP bi-directional prediction

<그림 3>은 식 (6)의 조건을 사용한 제안 알고리즘을 그림으로 나타낸 것이다.

단, 식 (6)은 L0와 L1의 참조 픽처가 동일한 경우 양방향 예측에 대한 고속화를 전혀 수행하지 않으므로, 이는 최적의 해결책이라고 볼 수 없다. 향후 이 부분에 대한 추가 연구를 수행할 계획이며, 제 5장에서는 식 (6)을 그대로 적용한 실험 결과를 보인다.

V. 실험 결과

1. 테스트 조건

제안한 방법의 효과를 검증하기 위해, HM15.0에 제안한 방법을 구현하고, 압축 성능 및 연산량 측면에서 비교를 수행하였다. 또한 HM15.0에 이미 구현되어 있는 고속화 알고리즘인 ESD 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 <표 1>의 영상들을 그대로 사용하였다.

2. 부호화 성능 및 복잡도

<표 7>은 식 (6)의 상수 α 값을 1.0에서 1.8까지 변화시켜 가면서 얻은 실험 결과이다. HM15.0에 대한 상대적인 결과 값으로 표시하였으며, AMVP의 양방향 예측을 완전히 끄는 경우와도 비교하기 위해 AMVP-NO-BI의 결과를 함께 포함시켰다.

<표 7>를 보면, $\alpha=1.0$ 을 사용하는 경우 5.2%의 부호화 복잡도를 감소시키고 압축 성능이 0.3% 하락하는 결과를

표 7. 제안한 AMVP 양방향 예측 고속화 방법의 압축 성능 및 복잡도
Table 7. Coding efficiency and encoding time of proposed fast encoding scheme for AMVP bi-directional prediction

	Class B	Class C	Class D	Average	Encoding time (%)
AMVP-NO-BI	5.7	3.6	4.3	4.5	90.7
$\alpha=1.0$	0.3	0.3	0.4	0.3	94.8
$\alpha=1.2$	0.5	0.5	0.5	0.5	92.5
$\alpha=1.4$	0.8	0.7	0.7	0.7	92.2
$\alpha=1.6$	0.9	0.7	0.8	0.8	91.6
$\alpha=1.8$	1.1	0.9	0.9	1.0	91.5

얻었다. AMVP의 양방향 예측을 전혀 사용하지 않는 경우 (AMVP-NO-BI) 얻을 수 있는 부호화 복잡도 감소가 9.3% 이므로, 이 중 55.9%를 감소시켰음을 알 수 있다. $\alpha=1.8$ 을 사용하는 경우, 8.5%의 부호화 복잡도를 감소시킬 수 있으며 압축 성능이 1.0% 하락한다. 8.5%의 부호화 복잡도 감소는 AMVP-NO-BI를 사용하였을 때 얻을 수 있는 부호화 복잡도 감소 대비 약 91.4%에 해당한다.

양방향 예측 모드의 성능/복잡도를 최적화하기 위해서는 제 3장에서 보인 것과 같이 AMVP의 양방향 예측 모드 뿐 아니라 총 merge candidate의 수를 감소시키는 것이 효과적이다. <표 8>은 $\alpha=1.0$, $\alpha=1.4$ 의 두 가지 경우에 대해 총 merge candidate를 4, 3, 2로 변화시켜가면서 성능 및 복잡도를 구해본 결과이다.

표 8. 총 merge candidate를 변화시키면서 측정된 제안한 방법의 압축 성능 및 복잡도

Table 8. Coding efficiency and encoding time of proposed method with respect to various number of merge candidates

α / MC	Class B	Class C	Class D	Average	Encoding time (%)
1.0 / 4	0.4	0.4	0.5	0.4	90.5
1.0 / 3	0.7	0.7	0.6	0.6	88.0
1.0 / 2	1.1	0.9	0.9	1.0	85.8
1.4 / 4	0.9	0.7	0.7	0.8	87.6
1.4 / 3	1.2	1.1	1.0	1.1	85.1
1.4 / 2	1.7	1.5	1.4	1.5	82.8

<표 8>과 <표 7>의 결과를 비교해 볼 때, 유사한 압축 성능 하락이 발생하는 경우에 대한 복잡도 감소가 총 merge candidate 수를 함께 조정하였을 때 더 크게 나타나는 것을 확인할 수 있었다. 예를 들어, 제안한 방법만을 사용한 경우, $\alpha=1.4$ 이면 압축 성능이 0.7% 하락하고 복잡도가 7.8% 감소하는데 비해, $\alpha=1.0$ 이고 총 merge candidate가 3인 경우 압축 성능은 유사하지만 부호화 복잡도는 12.0% 감소하였다. 또한 $\alpha=1.8$ 인 경우 압축 성능이 1.0% 하락하고 복잡도가 8.5% 감소하는데 비해, $\alpha=1.4$ 이고 총 merge candidate가 3인 경우 유사한 압축 성능에서 복잡도 감소폭이 14.9%로 상승하였다.

3. 복호화 복잡도

<표 9>는 복호화기 측에서 분석한 양방향 예측 모드의 출현 비율을 분석한 것이다. 일반적으로 양방향 예측 모드는 단방향 예측 모드에 비해 복호화기의 복잡도를 크게 증가시키므로, 이 비율이 낮아질수록 복호화기의 복잡도가 감소한다고 할 수 있다. 각 경우에 대한 압축 성능 변화도 함께 표시하였다.

표 9. 복호화기에서 구한 양방향 예측 모드 비율 및 압축 성능
Table 9. Ratio of bi-directional prediction and coding efficiency of proposed method

	Class B	Class C	Class D	Average (%)	BD-rates (%)
HM15.0	60.8	45.8	51.6	53.3	-
AMVP-NO-BI	55.8	38.8	44.8	47.2	4.5
제안한 방법 ($\alpha=1.0$ / MC=3)	55.2	40.5	45.2	47.0	0.6
제안한 방법 ($\alpha=1.0$ / MC=2)	47.6	34.9	42.1	41.5	1.0
제안한 방법 ($\alpha=1.4$ / MC=3)	52.9	35.2	42.5	43.5	1.1
제안한 방법 ($\alpha=1.4$ / MC=2)	44.0	28.6	37.5	36.7	1.5

<표 8>과 <표 9>의 결과로부터, 제안한 방법을 사용하면 압축 성능이 약 0.6%, 1.0%, 1.5% 하락할 때, 양방향 예측 모드의 비율을 각각 6.3%, 11.8%, 16.6% 감소시킬 수 있음을 알 수 있다.

4. 기존 고속 알고리즘과의 병행 사용 결과

<표 10>에서는 HM15.0에 포함되어 있는 고속화 기술인 ESD를 사용한 경우에도 제안한 기술의 효과가 유지되는지를 확인하기 위해서, ESD를 사용한 HM15.0의 결과 대비 제안한 기술의 성능/복잡도를 구해 보았다.

표 10. Early skip decision(ESD)을 사용한 경우 대비 제안한 방법의 압축 성능 및 복잡도
 Table 10. Coding efficiency and encoding time of proposed method with early skip decision

α / MC	Class B	Class C	Class D	Average	Encoding time (%)
1.0 / 2	1.0	0.9	0.8	0.9	84.7
1.4 / 2	1.9	1.5	1.3	1.6	80.7

<표 8>과 <표 10>을 비교해 볼 때, ESD를 사용하는 경우, 제안한 방법의 압축 성능 하락 폭은 거의 유사하지만 복잡도 감소는 소폭 커짐을 알 수 있다. 예를 들어 ESD를 사용하지 않는 경우 $\alpha=1.4$ 이고 총 merge candidate가 2인 경우 약 1.5%의 압축 성능 하락과 17.2%의 복잡도 감소를 얻었는데, ESD와 함께 사용한 경우 약 1.6%의 압축 성능 하락과 19.3%의 복잡도 감소를 얻었다. 이 결과로부터, 제안한 방법의 효과는 ESD와 독립적으로 유지된다고 판단할 수 있다.

VI. 결론

본 논문에서는 HEVC의 화면 간 예측 모드를 구성하는 각 요소들을 분석하고, 부호화 및 복호화 복잡도를 감소시키기 위한 양방향 예측 모드 부호화 조기 종료 조건 및 generalized P&B(GPB) 모드에서의 예외 조건, 그리고 최대 merge candidate수와의 병행 최적화를 제안하였다. 실험 결과, 제안한 방법을 사용하면 압축 성능 하락 폭 0.6%, 1.0%, 1.5%에서 부호화기 복잡도를 각각 12.0%, 14.2%, 17.2% 감소시킬 수 있었으며, 이 때 복호화기에서의 양방향 예측 모드 비율은 각각 6.3%, 11.8%, 16.6% 줄어든 것으로 나타

나 복호화 복잡도도 함께 감소하는 것을 확인할 수 있었다. 또한, 제안한 방법이 기존 HM15.0에 구현되어 있는 고속화 알고리즘과 함께 사용되는 경우에도 유사한 수준의 복잡도 감소 효과를 갖는 것을 보였다.

참고 문헌 (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] T. Wiegand, J.-R. Ohm, G. J. Sullivan, W. J. Han, R. Joshi, T. K. Tan and K. Ugur, "Special Section on the Joint Call for Proposals on High Efficiency Video Coding (HEVC) Standardization," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 20, No. 12, pp. 1661-1666, Dec. 2010.
- [6] 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.
- [7] 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.
- [8] K. Choi, S. H. Park, E. S. Jang, "Coding tree pruning based CU early termination," JCTVC-F092, 6th JCT-VC meeting, Jul. 2011, Torino, Italy.
- [9] J. Leng, L. Sun, T. Ikenaga and S. Sakaïda, "Content based hierarchical coding unit decision algorithm for HEVC," Proc. of Multimedia and Signal Processing (CMSP), pp. 56-59, May 2011.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] D. Wu, F. Pan, K. P. Lim and S. Wu, "Fast intermode decision in H.264/AVC video coding," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, No. 7, pp. 953-958, Jul. 2005.
- [14] Z. Huanqiang, C. Canhui and M. Kai-Kuang, "Fast mode decision for H.264/AVC based on macroblock motion activity," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 19, No. 4, pp. 491-499, Apr. 2009.
- [15] F. Bossen, "Common HM test conditions and software reference configurations," in Proc. of JCTVC-L1100, 12th JCT-VC meeting, Geneva, Switzerland, Jan. 2013.
- [16] HEVC reference software version 15.0 (HM15.0), <https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags/HM-15.0>, Jul. 2014.
- [17] G. Bjontegaard, "Calculation of average PSNR differences between RD curves," in Proc. VCEG-M33, 13th VCEG meeting, Austin, TX, USA, Apr. 2001.

저 자 소 개



한 우 진

- 1995년 2월 : KAIST 전산학과 공학사
- 1997년 2월 : KAIST 전산학과 공학석사
- 2002년 2월 : KAIST 전산학과 공학박사
- 2002년 3월 ~ 2003년 3월 : SL2 연구소장
- 2003년 4월 ~ 2011년 8월 : 삼성전자 DMC 연구소 수석연구원
- 2010년 10월 ~ 현재 : HEVC Working Draft Editor
- 2011년 9월 ~ 현재 : 가천대학교 소프트웨어설계경영학과 조교수
- ORCID : <http://orcid.org/0000-0001-5114-4117>
- 주관심분야 : 영상압축, 영상이해, 멀티미디어통신



안 준 형

- 2014년 2월 : 가천대학교 소프트웨어설계경영학과 학사
- 2014년 3월 ~ 현재 : 가천대학교 대학원 소프트웨어설계경영학과 석사과정
- ORCID : <http://orcid.org/0000-0002-4155-4865>
- 주관심분야 : 영상압축, 패턴인식



이 종 호

- 2011년 3월 ~ 현재 : 가천대학교 소프트웨어설계경영학과 학사과정
- ORCID : <http://orcid.org/0000-0002-2232-9849>
- 주관심분야 : 영상압축, 영상이해, 패턴인식, 빅데이터