HEVC 부호기의 Inter Prediction SAD 연산을 위한 효율적인 알고리즘

전성훈 · 류광기

한밭대학교 정보통신공학과

Efficient Computing Algorithm for Inter Prediction SAD of HEVC Encoder

Sung-Hun Jeon · Kwangki Ryoo

Dept. of Information & Communication Eng., Hanbat National University

E-mail : jsh4189@naver.com

요 약

본 논문에서는 고성능 HEVC 부호기를 위한 Inter Prediction SAD연산 구조의 효율적인 알고리즘을 제안한다. HEVC Inter Prediction에서의 Motion Estimation(ME)은 시간적 중복성을 제거하기 위하여 보간 된 참조 픽처에서 현재 PU와 상관도가 높은 예측 블록을 탐색하는 과정이다. ME는 전역 탐색(full search, FS) 알고리즘과 고속 탐색(fast search) 알고리즘을 이용한다. 전역 탐색 기법은 주어진 탐색 영역내의 모든 후보 블록에 대하여 움직임을 예측하기 때문에 최적의 결과를 보장하지만 연산량 및 연산시간이 많은 단점을 지닌다. 그러므로 본 논문에서는 Inter Prediction의 연산량 및 연산시간을 줄이기 위해 전역탐색에서 SAD연산을 재사용하여 연산 복잡도를 줄이는 새로운 알고리즘을 제안한다. 제안된 알고리즘은 HEVC 표준 소프트웨어 HM16.12에 적용하여 검증한 결과 기존 전역탐색 알고리즘보다 연산시간은 61%, BDBitrate는 11.81% 감소하였고, BDPSNR은 약0.5% 증가하였다.

ABSTRACT

In this paper, we propose an efficient algorithm for computing architecture for high-performance Inter Prediction SAD HEVC encoder. HEVC Motion Estimation (ME) of the Inter Prediction is a process for searching for the currently high prediction block PU and the correlation in the interpolated reference picture in order to remove temporal redundancy. ME algorithm uses full search(FS) or fast search algorithm. Full search technique has the guaranteed optimal results but has many disadvantages which include high calculation and operational time due to the motion prediction with respect to all candidate blocks in a given search area. Therefore, this paper proposes a new algorithm which reduces the computational complexity by reusing the SAD operation in full search to reduce the amount of calculation and computational time of the Inter Prediction. The proposed algorithm is applied to an HEVC standard software HM16.12. There was an improved operational time of 61% compared to the traditional full search algorithm, BDBitrate was decreased by 11.81% and BDPSNR increased by about 0.5%

키워드

HEVC, Inter Prediction, Motion Estimation, SAD, Full Search

Ⅰ. 서 론

HEVC는 H.264/AVC의 표준화를 수행하였던 ISO/IEC의 MPEG(Moving Picture Expert Group)과 ITU-T의 VCEG(Video Coding Expert Group)이 공동으로 개발한 차세대 동영상 압축 표준 기술이다.

HEVC에서는 CU(Coding Unit), PU(Prediction Unit), TU(Transform Unit) 3가지 종류의 부호화 단위 적용과 계층적 쿼드-트리(Quad-tree) 구조의 부호화 수행을 하며, 64x64부터 8x8까지 다양한 크기의 부호화 단위를 사용한다. 이외에도 화면 내 예측 방향의 증가, 향상된 움직임 예측 기법, 움직임 벡터 병합 및 세분화된 루프 필터를 적용

하여 H.264/AVC 대비 약 2배의 압축 성능을 보이지만 다양한 부호화 구조와 향상된 예측 기법에 따른 연산 복잡도가 H.264/AVC에 비해 크게 증가하였다[1].

새로운 기술들 중 움직임 예측은 현재 프레임과 가장 유사한 예측 프레임을 생성한다. 현재 PU와 참조 블록의 상관도를 비교하는 과정에서 정소화소와 부화소의 특징을 고려하여 간략화된 상관도 측정 방법인 SAD(Sum of Absolute Difference)를 사용한다. 하지만 화면간 예측의 Full-Search의 경우 4x8 PU부터 최대 64x64 크기까지 다양한 크기의 PU에 대한 SAD 연산을 수행하고 참조 픽처 탐색 영역에 대해 SAD 연산을 반복 수행하기 때문에 연산량 및 연산 시간이 높다.

본 논문에서는 고성능 HEVC 화면간 예측을 위해 반복 연산되는 SAD 연산의 연산량 및 연산 시간을 줄일 수 있는 새로운 알고리즘을 제안한다. 제안하는 알고리즘은 기존의 반복되는 SAD 연산의 결과 값을 최소 블록 크기인 4x4 단위로 SAD 값을 저장한다. 저장한 4x4블록 단위 SAD 결과 값은 분할되는 모든 블록 크기 SAD 연산에 재사용 하여 움직임 추정 후 화면간 예측을 수행한다.

본 논문의 구성은 다음과 같다. 2장에서는 HEVC의 표준 화면간 예측 기술에 대해 기술하고, 3장에서는 제안하는 화면간 예측 SAD 연산 알고리즘을 기술한다. 4장에서는 제안하는 알고리즘의 성능 비교를 기술하며, 마지막으로 5장에서는 본 연구의 결론을 도출한다.

Ⅱ. HEVC 화면간 예측

화면간 예측의 움직임추정(Motion Estimation ; ME)은 인코더에서 수행되는 과정으로 참조 픽처에서 현재 PU와 상관도가 높은 예측 블록을 탐색하는 과정이다. 움직임 추정 수행 결과 PU 단위로 참조 픽처 리스트의 정보, 참조 픽처 인덱스, 움직임 벡터와 차분신호를 변환 양자화 한 계수가 디코더로 전송된다. 디코더에서는 인코더로부터 전송된 주변 정보를 이용하여, 인코더와 동일한 예측 블록을 생성하고 양자화된 잔차 신호를 사용하여 복원 블록을 생성하는 움직임 보상 과정을 수행한다[2]. 그림 1 은 움직임 추정 과정을 나타낸다.

현재 PU와 참조 블록의 상관도를 비교하는 과정에서 간략화된 상관도 측정 방법인 SAD(Sum Absolute Different) 수식(1)을 사용한다.

$$SAD(i,j,k,l) = \sum |B_{cur}(i,j) - B_{ref}(k,l)| \quad (1)$$

B_{cur}는 현재 블록, B_{ref}는 참조 픽처 내에 존재하는 움직임 추정 후보 블록, i, j는 현재 PU의 위치 k, l은 움직임 추정 대상의 PU 위치를 나타낸다. 정수 화소의 움직임 추정은 현재 블록과 참조 블록의 차분값의 절대 값을 최소화하는 참조 블록을 선택함으로써 하나의 예측 블록을 선택한다.

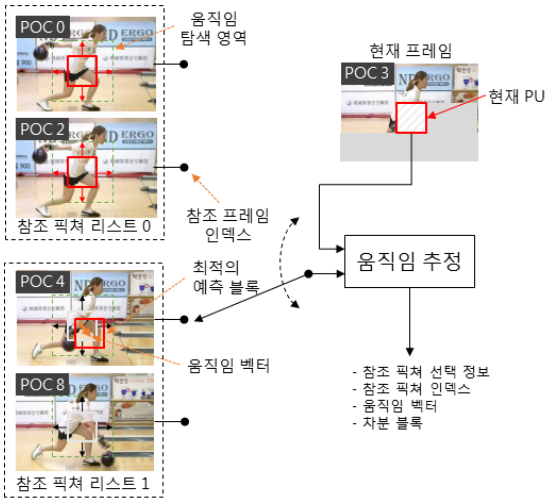

그림 1. 움직임 추정과정

화면간 예측에서 진행하는 PU 분할은 그림 2와 같이 여덟 가지의 분할 패턴 중 하나의 PU로 분할이 이루어진다. PART_NxN은 해당 CU가 최소 크기의 CU인 경우에만 분할이 가능하며 AMP(Asymmetric Motion Partition)라 불리는 PART_2NxnU, Part_2NxnD, Part_nLx2N, Part_nRx2N 형태의 분할은 코딩하려는 픽처가 최소 CU 크기의 CU인 경우에는 부호화 및 복호화 과정의 복잡도를 고려하여 사용하지 않는다.

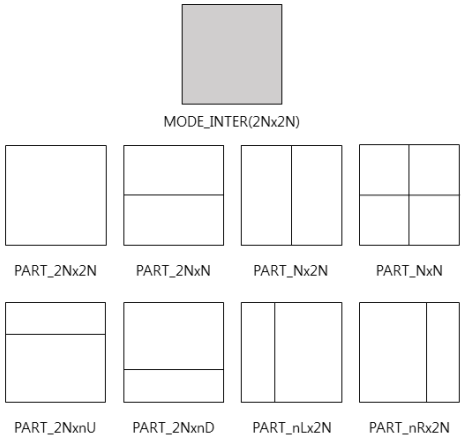

그림 2. 화면간 예측의 PU분할 모드

최적의 motion vector를 찾고자 할 때, search point의 수를 줄이는 3 step search (3SS), 4 step search (4SS), diamond search (DS) 등 많은 알고리즘이 제안되었으나 HEVC에서는 test zone search(TZS) 알고리즘과 Full-Search 알고리즘을 사용한다. TZS 알고리즘의 경우 단계별로 4개 혹은 8개씩의 search point들만 SAD 값을 계산하면서 최적의 위치를 탐색한다. Full-search 의 경우 그림 3과 같이 X, Y 픽셀 좌표 (-64, -64) ~ (64,

64)까지 탐색을 반복 수행하기 때문에 TZS에 비해 탐색 성능은 뛰어나지만 탐색범위가 커지면 연산량이 매우 커지는 단점이 있다.

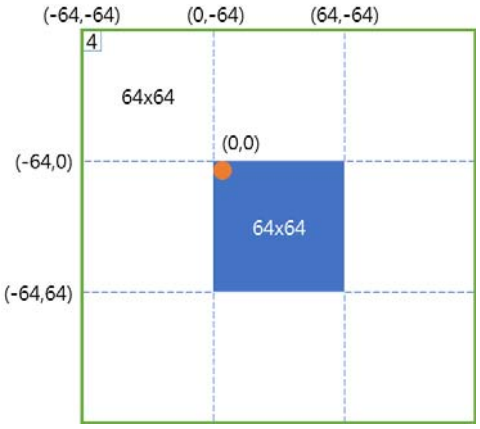

그림 3. 64x64 PU블록 Full-Search 탐색범위

Ⅲ. 제안하는 SAD연산 알고리즘

제안하는 SAD 연산 알고리즘은 Max CU 크기 32x32일 때, Full-Search 영역내의 4x4 블록단위로 SAD 연산 결과 값을 저장하고 모든 블록 분할 크기의 SAD 연산에 재사용하여 최적의 PU 분할 모드를 결정한다.

기존 Full-Search 에서의 SAD 연산 수행 과정은 움직임 탐색 영역 내에서 2Nx2N의 블록부터 Nx2N까지 모든 분할 블록에 대하여 SAD 값을 계산한다. 때문에 SAD 연산을 수행할 때, 반복 계산되는 탐색 영역이 지속적으로 발생하면서 연산의 중복성 및 복잡도가 높아 전체 인코딩 속도를 저하 시킨다. 표 1은 MaxCU 크기 32x32일 때 화면간 예측에서 갖는 PU 분할 모드를 나타낸다.

표 1. 화면간 예측 PU분할

PU SIZE	Depth		
	0	1	2
2Nx2N	32x32	16x16	8x8
2NxN	32x16	16x8	8x4
Nx2N	16x32	8x16	4x8
NxN	16x16	8x8	
2NxnU	32x8	16x4	
2NxnD	32x24	16x12	
nLx2N	8x32	4x16	
nRx2N	24x32	12x16	

이런 연산의 반복성을 줄이기 위해 32x32 블록 SAD 연산과정에서 4x4 블록단위로 SAD 값을 저장한다. 이후에는 4x4 블록단위로 저장된 SAD 결과 값을 이용하여 상위 크기 블록 SAD 연산에 재사용한다.

그림 3과 같이 4개의 4x4 블록의 SAD 결과 값의 합은 8x8블록의 SAD 값과 같으며, 이렇게 만들어진 4개의 8x8 블록의 결과값은 1개의 16x16 블록의 SAD 결과 값과 같다. 이와 같은 방법으로 4x4 단위 SAD 연산 결과 저장을 통하여 32x32 CU 내의 모든 17가지(표1) PU 분할에 대해 SAD 연산 결과값을 구할 수 있다.

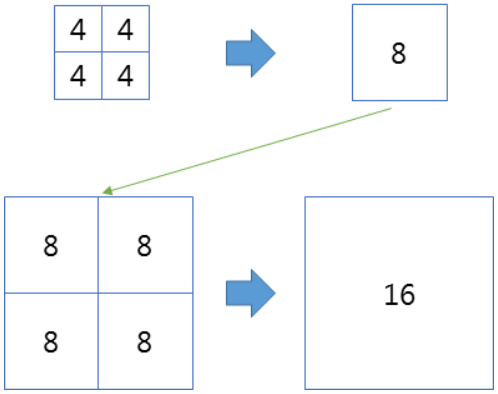

그림 4. 블록 재사용 방법

4x4 단위 SAD 연산 결과값을 효과적으로 재사용하기 위하여 32x32 크기의 PU에서 SAD 결과를 4x4 단위로 저장한 뒤 Index를 그림 5와 같이 부여한다. 이후 Index 값을 이용하여 재사용 한다. 예를 들어, 32x32 블록 SAD 연산 후 수행하게 되는 32x16 블록 SAD 연산과정에서는 저장한 4x4 블록 단위 SAD 연산결과 Index (1~32), (33~64)만을 호출해 32x32블록에서 나올 수 있는 32x16 SAD 결과값 2가지를 계산한다. 이후 이루어지는 모든 블록 분할에 대해서도 해당 영역의 Index만을 호출하여 4x4 SAD 연산결과를 재사용함으로써 같은 영역에서의 SAD 연산 중복률을 낮춘다.

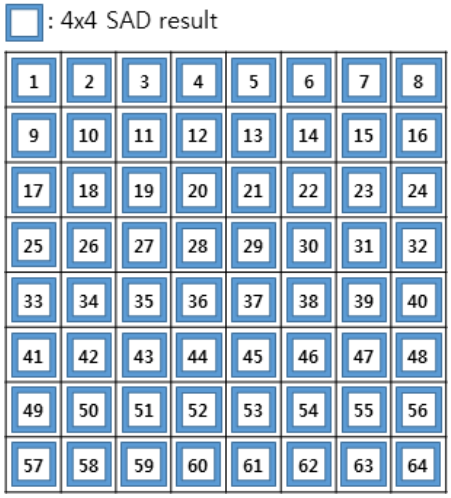

그림 5. 32x32 PU Index

표 2. HM-16.12 표준 소프트웨어와 제안하는 알고리즘 결과 비교

Sizes	Sequences	proposed			[3]		[4]	
		BDPSNR	BDBitrate	△TS(%)	BDBitrate	△TS(%)	BDBitrate	△TS(%)
Class A	Traffic	0.49	-12.26	60.5	1.88	60.2	1.67	45.3
	PeopleOnstreet	0.18	-3.76	59.3	1.11	41.2	1.44	29.2
Class B	Kimono	1.36	-30.02	60.4	1.93	53.2	1.04	36.7
	ParkScene	0.40	-11.34	62.3	2.09	55.2	0.86	44.4
	Cactus	0.31	-11.21	62.3	1.65	55.5	1.20	39.7
	BasketballDrive	1.51	-44.55	62.2	2.22	56.6	1.27	40.4
Class C	RaceHorsesC	1.16	-21.18	55.4	1.29	40.9	1.13	26.8
	BQMall	0.23	-4.66	52.8	1.70	52.6	1.27	42.2
	PartyScene	0.02	-0.26	57.3	2.47	40.3	0.99	31.2
	BasketballDrill	0.27	-5.98	54.6	0.96	48.3	1.58	37.2
Class D	RaceHorses	1.17	-18.70	70.2	1.46	30.3	0.97	23.0
	BlowingBubbles	0.01	-0.21	71.2	2.82	42.6	1.17	32.7
	BasketballPass	0.06	-1.14	60.0	3.30	53.1	0.84	40.7
	BQSquare	0.04	0.00	70.0	0.75	52.0	1.30	32.6
Average	-	0.515	-11.81	61.32	1.83	48.7	1.20	35.9

Ⅳ. 성능 비교

제안하는 SAD 알고리즘의 성능 향상을 비교하기 위해서 HM-16.12 표준 소프트웨어와 [3], [4]과의 성능 비교를 진행하였으며, 다양한 해상도에서 제안하는 알고리즘의 성능 향상을 비교하기 위해서 각 Class 영상을 QP 22, 27, 32, 37로 변경하여 테스트하였다. 표 2는 제안한 알고리즘에 대한 실험 결과를 보여준다. 제안하는 SAD 연산 알고리즘은 표준 소프트웨어 대비 평균적으로 61% 인코딩 속도가 향상되었으며 수식 (2)와 같이 계산하였다. 또한 BDPSNR은 0.5증가, BDBitrate는 11.81 감소하였다. [3], [4] 과의 성능 비교에서는 약 12%의 인코딩 속도가 향상되었다.

$$\triangle TS(\%) = \left(\frac{TS_{HM} - TS_{propose}}{TS_{HM}} \right) \times 100 \qquad (2)$$

Ⅴ. 결 론

제안하는 SAD연산 알고리즘은 HM-16.12 표준 소프트웨어와 성능 비교 결과 연산시간에서 크게 좋은 결과를 보였고 BDPSNR 및 BDBitrate에서도 좋은 결과를 보였다. 제안하는 알고리즘을 하드웨어 모듈로 설계시 연산시간을 최소화하는 장점이 있다. 하지만 4x4 SAD 연산 결과를 저장해야하는 메모리가 필요하기 때문에 효과적으로 메모리를 관리할 수 있는 연구가 필요하다.

감사의 글

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 해외ICT전문인력활용촉진사업(IITP-2015-R0134-16-1019)과 해외인재스카우팅사업(IITP-2016-R2418-16-0007)의 연구결과로 수행되었음

참고문헌

[1] G. Correa, P. Assuncao, L. Agostini, and L, Cruz, “Coding tree depth estimation for complexity reduction of HEVC,” in Data Compression Conference (DCC), 2013, 2013, pp. 43-52

[2] 심동규, 조현호, HEVC 표준 기술의 이해, pp. 177-179, 2014

[3] Jinlei Zhang, Bin Li, Houqiang Li, “An Efficeient Fast Mode Decision Method for Inter Prediction in HEVC,” in *IEEE* trans. Circuits, Syst. Video Technol., vol. 26, no.8, pp. 1502-1515, Aug. 2016

[4] S. Ahn, M. Kim, and S. Park, “Fast decision of CU partitioning based on SAO parameter, motion and PU/TU split information for HEVC,” in Proc. Picture Coding Symp., Dec. 2013, pp. 113-116.