

124-Mpixels/sec 하드웨어 HEVC 인코더의 화면간 예측 구현

조승현, 김현미, 이석호
한국전자통신연구원

shcho@etri.re.kr, chaos0218@etri.re.kr, shlee99@etri.re.kr

Implementation of Inter Prediction for 124-Mpixels/s Hardware HEVC Encoders

Seunghyun Cho Hyunmi Kim Sukho Lee

Electronics and Telecommunications Research Institute, Daejeon, Korea

요 약

본 논문에서는 하드웨어 HEVC 인코더의 화면간 예측의 구조를 제안한다. 제안된 화면간 예측은 정수화소 움직임 예측을 통해 코딩 트리 유닛 내 코딩 유닛 분할과 각 코딩 유닛의 분할모드를 결정한다. 그리고, 부화소 움직임 예측, Merge 모드 판단을 통해 예측 유닛의 움직임 벡터를 확정하고 움직임 보상을 수행한다. 이 과정에서 윌-왜곡 비용계산 및 보간 필터 등의 하드웨어 자원이 효율적으로 공유된다. 또한, 전력소모를 줄이기 위하여 코딩 유닛의 skip 여부의 조기 판단을 통해 부화소 움직임 예측 및 화면내-화면간 최종 예측모드 결정과정의 전부 또는 일부를 생략하는 방법을 제공한다. 제안된 화면간 예측을 포함한 하드웨어 HEVC 인코더를 구현하여 실험한 결과, 250 MHz 의 동작 주파수에서 초당 124 Mpixel 의 처리성능을 보였으며, HM-14.0 대비 PSNR 0.5~0.8 dB 수준의 화질열화를 나타냈다.

1. 서론

HEVC [1]는 ISO/IEC MPEG 과 ITU-T VCEG 에 의해 공동으로 제정된 차세대 비디오 압축 표준기술이며, 기존의 H.264/AVC 표준기술에 비해 압축효율이 두 배에 달하는 것으로 알려져 있다 [2]. H.264/AVC 가 고정된 크기의 매크로블록을 서브블록으로 분할하여 예측하는 것과는 달리, HEVC 는 코딩 구조에 변화를 주어 보다 큰 사이즈의 블록에 대한 예측을 효과적이고 유연하게 지원할 수 있도록 하였다. 따라서, HEVC 에서 화면간 예측에 사용 가능한 예측 유닛(Prediction Unit, PU)의 크기도 더욱 커졌을 뿐만 아니라 비대칭 움직임 분할(Asymmetric Motion Partition, AMP)을 지원함으로써 PU 의 형태가 다양해져 보다 정교한 예측이 가능하게 되었다. 또한, 움직임 데이터의 코딩 방식도 기존과 같은 단일 유도가 아닌 인덱싱 방식으로 보다 정교해졌다 [2]. 이러한 변화들은 효율적인 HEVC 인코더를 구현하기 위해 화면간 예측 과정에 기존보다 많은 계산량이 요구되며, 하드웨어로 구현 시에는 인코더 칩 면적 증가의 원인이 될 수 있음을 의미한다.

일반적으로, 하드웨어 기반의 비디오 인코더에서는 인코딩의 각 과정을 모듈 별로 구현한 파이프라인을 통해 기존의 매크로블록과 같이 코딩 구조의 기본 단위가 되는 블록을 각 모듈에서 병렬처리한다. 본 논문에서는 그림 1 에

나타낸 코딩 트리 유닛(Coding Tree Unit, CTU) 단위 파이프라인 구조를 갖는 하드웨어 HEVC 인코더를 위한 효율적인 화면간 예측의 구조에 대해 제안한다. 그림 1 의 상단에 나타낸 것처럼, 제안하는 화면간 예측은 CTU 단위 인코더 파이프라인에서 IME 와 PMR 의 두 단계에 걸쳐 수행되며, 그림 1 에 회색 상자로 나타낸 정수화소 움직임 예측(Integer-pel Motion Estimation, IME) 모듈과 부화소 움직임 예측(Fractional-pel Motion Estimation, FME) 모듈, SKIP 모듈, 색차신호 움직임 보상(Motion Compensation, MC) 등 복수의 모듈들로 구현된다. 각 모듈들은 코딩 효율과 처리 속도를 크게 떨어뜨리지 않으면서도 복잡도를 낮추기 위해 하드웨어 자원을 공유하도록 설계되었다.

본 논문의 구성은 다음과 같다. 2 절에서는 IME 모듈에 대해 살펴본 후, 3 절에서는 FME, SKIP, Chroma MC 모듈에

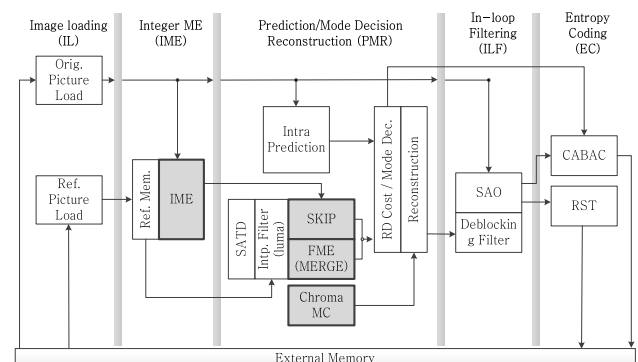


그림 1. 하드웨어 HEVC 인코더의 CTU 단위 파이프라인 구조의 예

대해 살펴본다. 4 절에서 제안된 움직임 예측 구조의 성능을 실험을 통해서 확인한 후, 마지막으로 5 절에서 본 논문에 대한 결론을 맺는다.

2. 정수화소 움직임 예측 모듈

제안하는 HEVC 화면간 예측의 첫 번째 단계는 IME 단계로, 그림 1 의 IL 단계에서 로딩된 참조픽처에 대해 정수화소 움직임 예측을 수행하여 CTU 내 CU 분할과 CU 의 분할모드, 즉 *PartMode* 를 결정한다. 그림 2 에 IME 모듈의 탐색 영역 결정 및 SAD 계산에 대해 나타냈다. IME 모듈의

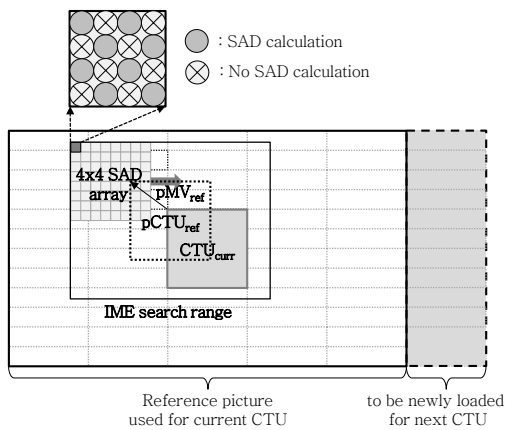


그림 2. IME 모듈의 탐색 영역 결정 및 SAD 계산

탐색영역은 그림 2 의 참조픽처 상에 CTU_{curr} 로 나타난 현재 CTU 의 위치로부터 pMV_{ref} 만큼 떨어진 곳에 위치한 가상의 $pCTU_{ref}$ 를 중심으로 정한다. IME 탐색 영역의 크기는 인코더에서 지원하는 CTU 의 크기, 탐색방법, 화면크기, 프레임율, 동작 주파수 등을 고려하여 정의되어야 하는데, 본 논문의 구현에서는 최대 32×32 CTU, 순차탐색(full search), 1920×1080 픽셀, 60 fps 프레임율, 250 MHz 동작 주파수를 가정하여 가로 및 세로 방향으로 각각 ± 24 , ± 16 의 탐색영역을 갖도록 하였다. pMV_{ref} 는 주변 PU 의 움직임 데이터 정보를 참조하여 그들의 중간 값을 취해 계산한다. IME 의 각 탐색 지점에서 각 PU 의 예측 비용은

$$J_{IME} = SAD + \lambda_{IME} \cdot B_{IME} \quad (\text{수식 1})$$

와 같이 계산하며 Sum of Absolute Differences(SAD) 계산에 필요한 하드웨어 크기를 줄이기 위해 그림 2 에 나타난 것처럼 4×4 단위로 SAD 를 계산할 때 일부의 샘플만을 사용하도록 한다. 수식 1 에서 λ_{IME} 는 HM [5]의 λ_{pred} 값을 참고하여 쉬프트와 덧셈 연산으로 구현 가능한 근사치를 구해 사용하며 B_{IME} 는 pMV_{ref} 에서 각 탐색 지점으로 향하는 움직임 벡터를 표현하는데 필요한 비트 수를 나타낸다. IME 탐색이 끝난 후, 각 PU 별로 저장된 최적의 예측 비용과 움직임 벡터를 바탕으로 32×32 CTU 내 CU 분할과 각 CU 의 *PartMode* 를 결정한다.

3. 부화소 움직임 예측, SKIP, 색차신호 움직임 보상 모듈

제안하는 화면간 예측의 두 번째 단계는 그림 1 에 나타난 하드웨어 HEVC 인코더 파이프라인의 PMR 단계에서 수행된다. IME 단계와 달리 PMR 단계에서는 주변 PU 의 움직임 데이터를 정확히 알 수 있기 때문에, skip 을 포함한 Merge 모드 판단, 최종 전송될 움직임 데이터 결정, 화면내-화면간 예측모드 판단 등이 모두 PMR 단계에서 수행된다. 그림 1 에 SKIP 으로 표기한 모듈은 IME 단계에서 결정된 각 CU 에 대해 Merge Candidate List(MCL) 상의 Merge 후보 중 CU skip 결정을 위해 사용될 최적의 후보 하나를 선택한다. 한편, 그림 1 의 FME 모듈은 IME 단계에서 결정된 각 CU 내 PU 의 움직임 벡터 주변의 부화소를 탐색하여 해당 PU 의 예측을 위한 움직임 벡터를 최종 결정한다. 이때, 탐색할 부화소 위치가 MCL 상의 후보들이 가리키는 위치 중 하나와 동일할 경우 Merge 모드를 이용한 움직임 벡터 예측이 고려된다. 그렇지 않은 경우에는 Advanced Motion Vector Prediction (AMVP)을 위한 후보 움직임 벡터 중 현재 부화소 위치와 가까운 것을 선택하여 움직임 벡터 예측에 사용한다.

그림 1 에서 볼 수 있는 것처럼, SKIP 모듈과 FME 모듈은 앞서 보인 수식 1 에서 SAD 대신 Sum of Absolute Transformed Differences(SATD)를 사용하여 각 MCL 후보 또는 부화소 위치에 대한 예측 비용을 계산하기 위해 8×8 단위 SATD 계산 하드웨어를 공유한다. 아울러, 두 모듈은 부화소 위치에 대한 샘플 값을 생성하기 위해 Luma 샘플 보간 필터를 공유한다. 더 나아가서, SKIP 모듈과 FME 모듈은 화면내-화면간 최종 예측모드 결정을 위한 RD cost 계산 모듈을 공유함으로써 하드웨어 비용의 증가를 억제할 수 있다. 그림 3 에 SKIP 모듈과 FME 모듈의 하드웨어 자원 공유를 위한 스케줄링의 예로 16×16 CU 의 PMR 단계 수행과정을 그림으로 나타냈다. 그림 3 에서, 하나의 16×16 CU 에 대하여 SKIP 과 FME 모듈을 순차적으로 동작시킴으로써 하드웨어 자원을 공유한다. SKIP 과 FME 모듈은 모두 Luma 샘플을 사용한

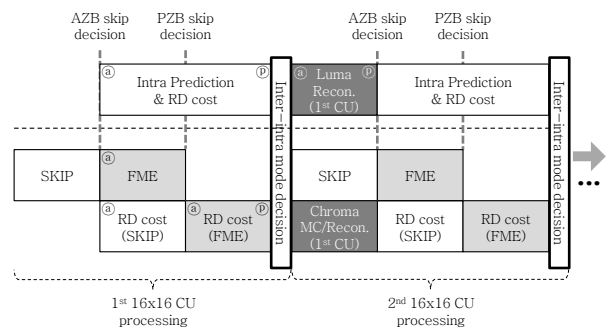


그림 3. 16×16 CU 의 PMR 단계 수행과정

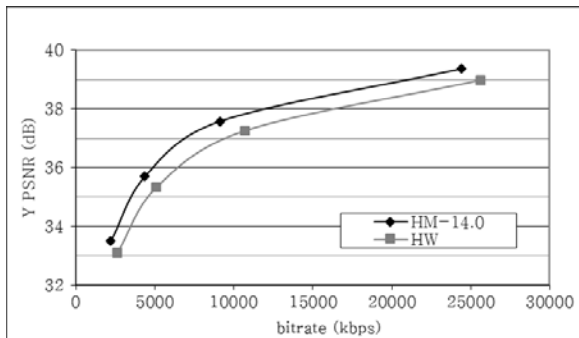
화면간 예측을 수행하며, 화면내-화면간 최종 예측모드 결정 후에 그림 3 에 진한 회색으로 나타난 것처럼 Chroma 샘플에 대한 MC, Luma 및 Chroma 샘플에 대한 복원이 필요하다.

제안된 화면간 예측의 구조에는 그림 3 의 SKIP 모듈에서 CU 의 skip 여부를 초기에 판단하여 해당 CU 에 대한 나머지 처리과정을 생략하여 전력소모를 줄이는 방법이 사용된다. CU 의 skip 여부의 초기판단은 SATD 기반의 All Zero Block(AZB) [3] 검출을 통해 이루어지며, AZB 로 검출되지 않은 경우에는 RD cost 계산을 위한 변환 및 양자화 후에 정상적인 CU skip 여부 판단과 더불어 Pseudo Zero Block(PZB) [4] 검출을 통해 추가적으로 CU skip 여부를 판단한다. AZB 와 PZB 검출에 의한 CU skip 의 판단 가능 시점을 그림 3 의

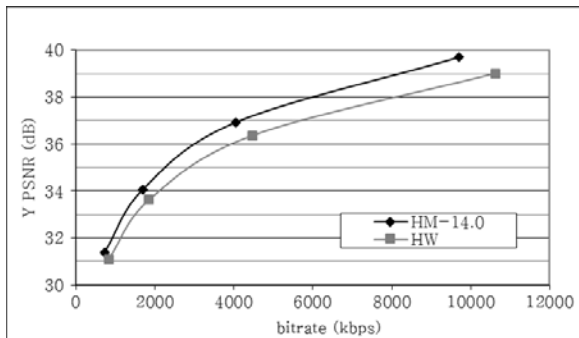
상단에 나타났다. 그림 3 에서 ㉔는 첫 번째 16×16 CU 가 AZB 에 의한 skip 으로 결정된 경우, ㉕는 첫 번째 16×16 CU 가 PZB 에 의한 skip 으로 결정된 경우에 전부 또는 일부의 수행과정이 생략될 수 있는 모듈들을 나타낸다.

4. 실험결과

제안된 움직임 예측 구조의 효율성을 검토하기 위해, 그림 1 의 파이프라인 구조를 갖고 2 절과 3 절에서 논의된 모듈들을 모두 포함하는 하드웨어 HEVC 인코더를 설계하여 HM 과 비교실험을 진행하였다. 그림 4 에 HM-14.0 [5] 과 구현된 하드웨어 HEVC 인코더의 *BasketballDrive* 와 *ParkScene* 영상에 대한 비트율-PSNR 비교를 그래프로 나타냈다. 구현된 인코더는 HM-14.0 과 비교했을 때 동일 비트율에서 약 0.5~0.8 dB 의 화질열화를 보였다. 실험을 통해 측정된 화질열화가 화면간 예측뿐만 아니라 하드웨어 HEVC 인코더 전체의 화질열화에 유의해야 한다.



(a)



(b)

그림 4. HM-14.0 과 구현된 하드웨어 HEVC 인코더의 비트율-PSNR 비교: (a) *BasketballDrive*; (b) *ParkScene*

제안된 화면간 예측을 구성하는 모듈들은 모두 Verilog HDL 을 사용하여 구성되었으며, 약 250 MHz 의 동작 주파수에서 124-Mpixels/sec 의 처리성능을 나타내어 Full-HD 60fps 영상의 실시간 처리가 가능한 것으로 판단되었다.

5. 결론

본 논문을 통해 하드웨어 HEVC 인코더를 위한 화면간 예측의 구조를 제안하였다. 제안된 화면간 예측은 두 단계의

하드웨어 HEVC 인코더 파이프라인에 걸쳐 수행되는데, 그 중 첫 번째 단계에서는 정수화소 움직임 예측을 통해 CTU 내 CU 분할과 각 CU 의 *PartMode* 를 결정한다. 두 번째 단계에서는 FME, CU skip 판단, MC, 복원 등의 과정을 수행한다. 제안된 화면간 예측을 구성하는 모듈간에 하드웨어 자원을 공유하기 위한 방법과 전력소모 감소를 위한 방법들도 함께 제안되었다. 제안된 화면간 예측을 포함한 하드웨어 HEVC 인코더는 초당 124 Mpixel 의 처리성능을 보였으며, 동일 비트율에서 HM-14.0 대비 PSNR 0.5~0.8 dB 수준의 화질열화를 나타냈다.

참고문헌

- [1] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, *High Efficiency Video Coding (HEVC) Text Specification Draft 10*, document JCTVC-L1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Mar. 2013.
- [2] G. J. Sullivan, J.-R. Ohm, W.-J. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. Circuits Systems for Video Technol.*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.
- [3] H. Wang and S. Kwong, "Prediction of Zero Quantized DCT Coefficients in H.264/AVC Using Hadamard Transformed Information", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 4, April. 2008
- [4] K. Lee, H.-J. Lee, J. Kim and Y. Choi, "A Novel Algorithm for Zero Block Detection in High Efficiency Video Coding", *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, Dec. 2013
- [5] HM Reference Software 14.0 [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware