SIMD를 이용한 HEVC 하다마드 트랜스폼의 고속 구현

*유종훈 **조현호 ***심동규

광운대학교

*mmyjh@kw.ac.kr

Fast implementation of Hadamard transformation of HEVC with SIMD

*Jong-Hun You **Hyun-Ho Jo ***Dong-Gyu Sim

Kwangwoon University

요약

본 논문에서는 SIMD(Single Instruction Multiple Data) 프로세서를 사용한 HEVC 부호화기의 하다마트 트랜스폼 고속화를 제안한다. 본 논문에서는 MMX와 SSE 레지스터를 사용하여 하다마드 트랜스폼을 SIMD 연산으로 대체함으로써 메모리 접근 횟수와 명령어의 수를 줄여 부호화기를 고속화 하였다. 또한, HEVC의 10비트 입력에 따른 SIMD 구조의 비효율적인 구현을 해결하기 위하여 하다마드 트랜스폼의 입력 픽셀 비트수를 감소시키는 IBDD(Internal Bit Depth Decreasing)를 제안했다. HEVC 부호화기에 하다마드 트랜스폼을 SIMD 연산으로 대체한 결과 부호화 효율의 저하 없이, 부호화기의 수행 시간은 10% 감소되었다.

1. 서론

MPEG(Moving Picture Experts Group)과 VCEG(Video Coding Experts Group)은 JCT-VC(Joint Collaborative Team on Video Coding)를 구성하고, HEVC(High Efficiency Video Coding)라는 차세대 비디오 압축 표준을 개발하고 있다.[1] HEVC는 H.264/AVC와 동일 화질 대비 50%의 비트율 감소, HD(High Definition), UD(Ultra Definition)급과 같은 고해상도의 영상을 목표로 표준화가 시작되었다.

HEVC는 기본 부호화 단위로 CTB(Coding Tree Block)를 사용하여 64×64에서 4×4까지 트리 구조로 영상 데이터를 표현한다. 또한, CU(Coding Unit), PU(Prediction Unit), TU(Transform Unit)라는 세부적인 블록 파티션 기술을 사용하여 독립적인 깊이 맵을 구성한다. 선택된 유닛에 대해 34개의 방향성을 이용하여 세밀한 예측을 하는 화면 내 예측, AMVP(Advanced Motion Vector Prediction)를 사용하는 화면 간 예측에서 H.264/AVC와 비교하여 동일 비트율 대비 향상된 화질, 동일 화질 대비 높은 부호화 성능을 얻는다. 이외에도 SAO(Sample Adaptive Offset), ALF(Adaptive Loop Filter), Merge 등의 부호화 성능을 향상 시키는 기술이 존재한다.[2][3]

하지만 HEVC의 부호화 효율 향상 이면에는 복잡도 문제가 존재한다. 실시간 HEVC 부호화기를 구현하기 위해서는 HEVC에서 부호화기의 복잡도가 높은 모듈인 화면 간 예측에 대하여 최적화 할 필요가 있다. HEVC의 화면 간 예측은 GPB(Generalized P and B), 다중 참조 프레임, RDO(Rate Distortion Optimization)에 따라 복잡도가 크게 증가한다. 따라서 화면 간 예측 모드를 최적화하기 위해서는 이러한 기술들에 대하여 알고리즘 레벨에서 최적화를 수행하는 것이 필요하다. 이때, 부호화기의 성능의 저하 없이 부호화기의 속도를 최적화 하는 것이 매우 중요하다. 본 논문에서는 실시간 부호화기를 구현하기 위하여, 이러한 알고리즘 레벨과 별개로 동시에 적용될 수 있는 명령어 수준의 병렬성을 사용한 최적화를 소개한다. 특히, 본 논문에서는 화면 간 예측 모드에서 높은 복잡도를 갖는 하다마트 트랜스폼에 대하여 SIMD 프로세스를 사용하여 복잡도를 감소시켰으며, 이때 SIMD 구조에 적합하도록 하다마트 트랜스폼의 입력 값을 조절하는 IBDD를 제안함으로써 추가적으로 부호화 속도를 향상 시켰다.[4]

본 논문의 구성은 다음과 같다. 2장에서는 부호화기의 복잡도 분석 및 SIMD 프로세서를 사용한 HEVC 부호화기의 고속화 방법을 개발한다. 3장에서는 고속화 결과와 성능 측정 결과를 논의하고 4장에서는 본 논문에 대한 결론 및 향후 연구의 진행 방향을 제시하고 본 논문을 마치도록 한다.

2. SIMD를 사용한 HEVC 부호화기 고속화

가. 부호화기 복잡도 분석

HEVC 부호화기의 복잡도는 화면 간 예측이 80%,

화면 내 예측이 3~5% 정도를 차지한다. 화면 간 예측 내에서는 하다마드 트랜스폼이 25%, 인터폴레이션이 30%, 움직임 보상 20% 차지한다. 세 모듈 중 하다마드 트랜스폼은 함수 호출 빈도가 높으며 화면 간, 화면 내 예측에서 반복적으로 사용 된다. 또한, SIMD 프로세서를 이용하기에 적합한 구조로 되어있다. 표 1은 BasketBallDrill 영상에 대한 하다마드 트랜스폼이 차지하는 복잡도 비중을 나타낸다.

표 1. BasketBallDrill 하다마드 트랜스폼 차지 비중

함 수 명	Low Delay	Random Access
xCalcHADs4×4	1.86%	1.79%
xCalcHADs8×8	16.45%	16.77%

나. SIMD를 사용한 하다마드 트랜스폼 고속화

하다마드 트랜스폼은 원본 영상과 리컨 영상의 차분 값을 입력으로, 계산량이 적은 덧셈 및 뺄셈 연산으로 트랜스폼 도메인의 계수 값을 얻는다.[5] 트랜스폼 계수 값 한 개를 얻기까지 사용되는 연산의 수는 2차원 4×4 트랜스폼은 4회, 8×8 트랜스폼은 6회이다. 그림 1은 4x4 하다마드 트랜스폼의 예로써 16개의 트랜스폼 계수를 얻기 위해 그림 1과 동일한 과정을 4회 반복해야 한다. 그러나 이와 같은 바이트 단위의 메모리 접근은 외부 메모리의 접근 횟수를 증가시키기 때문에 부호화 속도를 저하 시키게 된다.

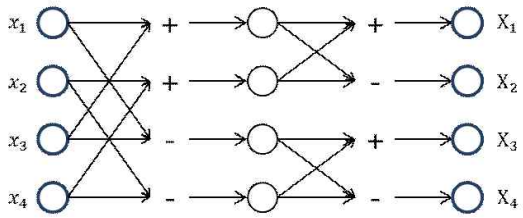

그림 1. 4×4 하다마드 트랜스폼

동일한 계산 알고리즘이 적용되는 같은 행의 픽셀 값들을 MMX, SSE 레지스터를 사용 하면 연산 횟수를 줄일 수 있을 뿐만 아니라 메모리 접근 횟수도 줄일 수 있다. 그림 2는 동일한 연산을 수행하는 그림 1의 픽셀들을 그룹화 하여, 64비트 MMX 레지스터를 사용하여 고속화 시키는 예이다.

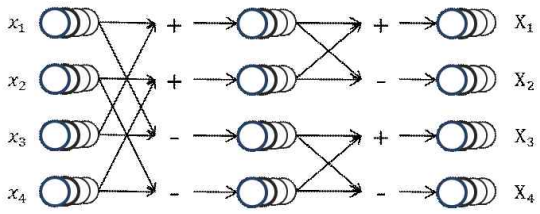

그림 2. MMX를 사용한 4×4 하다마드 트랜스폼

HEVC는 IBDI(Internal Bit Depth Increasing) 기술을 사용하여, 내부 계산 과정에서 픽셀 당 10비트를 사용한다. 2차원 8×8 하다마드 트랜스폼은 차분 값에 6회의 연산을 추가로, 총 7회의 덧셈, 뺄셈 과정을 수행하게 된다. 계산 결과 17비트의 결과 값이 발생 할 수 있으며, SSE 레지스터를 사용하는 경우, 각 픽셀 값을 더하는 과정에서 20비트 결과가 생성 될 수 있다. 그림 3은 8×8 하다마드 트랜스폼을 128비트 SSE 레지스터에 8개의 데이터를 처리 하는 과정이다. 두 번째 연산에서 오버플로우를 방지하기 위해 SSE 레지스터에 4개의 데이터를 사용하면 연산 횟수가 늘어난다..

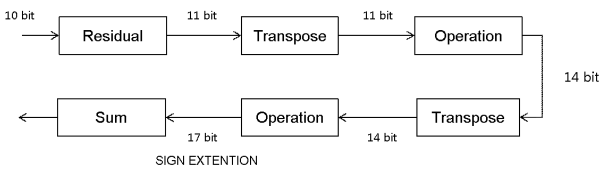

그림 3. SSE 레지스터 사용한 하다마드 트랜스폼 과정

따라서 본 논문에서는 하다마드 트랜스폼 고속화의 효율을 개선하기 위하여 SIMD 인스트럭션의 수를 최소화하기 위한 IBDD를 제안한다. 제안하는 IBDD는 영상의 차분 값의 표현범위를 축소하여 부호 확장에 들어가는 명령어를 최소화 하도록 한다. 그림 4는 제안하는 알고리즘이 적용되는 위치를 나타낸다. 두 번째 연산이 끝나고 레지스터 내의 픽셀 값들을 더하는 과정에서 부호 확장을 한다.

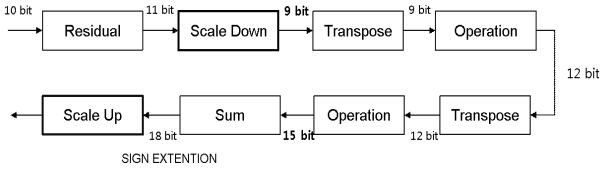

그림 4. IBDD, SEE 레지스터를 사용한
하다마드 트랜스폼에 수행과정

IBDD를 사용하지 않을 경우, 사용되는 SIMD 명령어의 개수는 194개, 1비트 시프트 하였을 경우 사용되는 명령어의 개수는 188개, 2비트 시프트 하였을 경우 사용되는 명령어의 개수는 158개로 하다마드의 최대 18%, 전체 부호화기의 2~3%의 성능 향상이 기대된다.

3장 실험결과 및 분석

본 논문에서는 제안하는 IBDD를 적용한 하다마드 트랜스폼의 고속화 알고리즘의 결과를 확인하기 위하여 HM 4.0 참조 소프트웨어를 사용하였다. 실험 환경은 Microsoft Window 7 Ultimate K(64비트) 기반의 Intel Core 2 Duo 프로세서를 사용 하였다. 실험 영상으로는 C class의 BasketballDrive를 사용하였고 random access, high efficiency로 부호화 하였다.

제안하는 IBDD를 적용한 하다마드 트랜스폼의 고속화 실험 결과는 표 2와 같다. 하다마드 트랜스폼을 SIMD 형태로 구현할 때 IBDD를 제안한 경우 부호화 효율의 저하가 거의 없이 부호화기 전체 수행 시간을 약 10% 감소시킨 것을 확인하였다.

표 2. 고속화 수행 시간 및 객관적 화질 비교

구분	QP	Bitrate	'PSNR	수행시간	감소비율
Original	22	3491.66	40.8075	3116.245	
	27	1687.372	37.7026	2744.425	
	32	833.052	34.807	2466.114	
	37	443.152	32.3296	2276.353	
Proposed	22	3487.636	40.8057	2805.134	9.98%
	27	1686.06	37.7013	2444.118	10.94%
	32	831.144	34.7969	2169.558	12.03%
	37	444.764	32.3281	1994.143	12.40%

4장 결론

본 논문에서는 HEVC 부호화기에서 높은 계산량을 갖는 하다마드 트랜스폼에 대하여 SIMD 연산을 적용하였다. 또한, 효과적인 SIMD 연산의 적용을 위하여 IBDD를 제안하였다. IBDD를 적용하여 하다마드 트랜스폼을 SIMD 연산으로 구현한 결과 부호화 성능의 저하 없이, 부호화기의 수행 시간을 약 10% 감소 시켰다. 추후에는 IF(Interpolation Filter)와 DCT(Discrete Cosine Transform)에 대하여 SIMD 연산을 사용한 최적화 연구를 수행할 것이다.

감사의 글

"본 연구는 지식경제부 및 한국산업기술평가관리원의 산업원천기술개발사업(정보통신)의 일환으로 수행하였음." [10039199, 인지품질 기반 스케일러블 3D 비디오 코덱 핵심 기술 연구]

참고 문헌

[1] T. Wiegand, W-J. Han, B. Bross, J-R. Ohm, G. J. Sulivan, "WD3: Working Draft 3 of High-Efficiency Video Coding," *JCTVC-E603, Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 5th Meeting:* Geneva, CH, 16-23 March, 2011

[2] M. Guo, X. Guo, and S. Lie, "Improved Intra Mode Coding," *JCTVC-D166,* Daegu, Korea, Jan., 2011

[3] I-K. Kim, T. Lee, "Improved motion vector predictor selection in AMVP," *JCTVC-D337, Joint Collaborative team on Video Coding of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11,* Daegu, Korea, Jan., 2011

[4] X. Li, F. Wei, "An improved practical efficient implementation of ICT used in H.264," *IEEE International Multimedia and Expo,* vol. 2, pp.1163-1166, June. 2004

[5] C. Zhu and B. Xiong, "Transform-Exempted Calculation of Sum of Absolute Hadamard Transformed Differences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 8, pp.1183-1188, Aug. 2009.