AVX2 명령어를 이용한 HEVC 역 이산여현변환 고속화

김우리, 조현호, 안용조, 심동규
광운대학교
{wrkim, idjhh, madein1st, dgsim}@kw.ac.kr

Fast implementation of HEVC inverse DCT using AVX2 instructions

Woori Kim, Hyunho Jo, Yong-Jo Ahn, and Dong-Gyu Sim
Kwangwoon University

요 약

본 논문에서는 HEVC (High Efficiency Video Coding)의 IDCT (Inverse Discrete Cosine Transform) 모듈을 AVX2 (Advanced Vector Extensions 2) 명령어 셋을 사용하여 고속화하는 방법을 제안한다. 제안하는 방법은 4 개의 4×4 블록을 AVX2 레지스터에 로드 한 후, 동시에 AVX2 명령어 셋을 통해 한 번에 IDCT 를 수행한다. 제안하는 방법은 4×4 블록 단위로 순차적으로 SIMD (Single Instruction Multiple Data) 명령어 셋을 통해 IDCT 를 수행하는 방법에 비해 명령어 단위의 병렬화 성능을 극대화한다. 실험 결과, HEVC 디코더의 4×4 IDCT 에 SIMD 명령어 셋을 적용한 경우 기존의 HM-12.1 에 비해 평균 3.35 배 수행 속도를 향상 시킨 반면, 제안하는 방법은 HM-12.1 에 비해 평균 9.50 배 수행 속도를 향상 시켰다.

1. 서론

최근 고품질, 고해상도 콘텐츠에 대한 수요가 증가함에 따라, ISO/IEC MPEG (Moving Picture Experts Group)과 ITU-T VCEG (Video Coding Experts Group)은 JCT-VC (Joint Collaborative Team on Video Coding)을 결성하여 새로운 비디오 압축 기술인 HEVC (High Efficiency Video Coding)를 표준화하였다[1][2]. HEVC 는 다양한 크기의 코딩, 변환, 예측 블록 및 향상된 예측 부호화 기술을 사용함으로써 H.264/AVC 에 비해 약 50% 높은 압축 성능을 갖지만, 부호화기 및 복호화기의 복잡도가 상대적으로 높다[3]. 특히, 복호화기에서는 8-tap 보간 필터, 큰 크기의 변환 커널, 두 종류의 인-루프 필터의 사용이 복잡도 증가의 주요 요인이다. 따라서 HEVC 실시간 복호화기를 위해서는 이러한 모듈들에 대한 최적화가 연구가 필요하다.

실시간 HEVC 부호화기 및 복호화기를 위한 HEVC 최적화 연구로서, 최근 HEVC 부호화기 및 복호화기의 일부 모듈을 SIMD (Single Instruction Multiple Data) 명령어 셋을 사용하여 복잡도를 감소시키는 연구들이 진행되었다[4]. 그러나 HEVC IDCT (Inverse Discrete Cosine Transform) 모듈의 4×4 커널은 256 비트 크기의 레지스터를 충분히 사용하지 못하여, 시스템의 성능을 극대화 시키지 못한다는 문제점이 있다. 본 논문에서는 HEVC 복호화기의 IDCT 모듈에 SIMD 명령어 셋을 적용할 때 인접하는 4 개의 블록을 동시에 처리함으로써 256 비트 레지스터를 최대한 사용하고, 이를 통해 명령어 레벨의 병렬화를 극대화 시킬 수 있는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2 장에서는 기존의 연구 방법 및 문제점들에 대해 소개한다. 3 장에서는 본 논문에서 제안하는 기법을 설명하고, 4 장에서는 제안한 알고리즘을 성능을 평가 및 분석한다. 마지막으로 5 장에서는 결론 및 향후 연구 방향을 소개한다.

2. SIMD 명령어를 통한 IDCT 고속화

HEVC 복호화기의 역변환 과정에는 IDCT 기반의 4×4, 8×8, 16×16, 32×32 크기의 네 종류의 커널과 IDST (Inverse Discrete Sine Transform) 기반의 4×4 크기의 커널이 사용된다[5]. HEVC 의 역변환 모듈은 기존의 비디오 압축 표준 기술과 유사하게 partial butterfly 알고리즘을 통해 최적화 될 수 있다. 그러나 SIMD 기반의 최적화를 수행하는 경우에는 동시에 여러 데이터에 연산의 적용이 용이한 행렬 곱 형태의 구조가 보다 적합하다.

본 논문에서는 SIMD 명령어를 사용하여 HEVC 의 IDCT 모듈을 최적화 하는 경우의 성능을 확인하기 위하여 행렬 곱 구조에 기반하여 SSE(Streaming SIMD Extention) 명령어 셋을 통해 IDCT 모듈의 최적화를 수행하였다. 표 1 은 HEVC 참조 소프트웨어인 HM-

12.1 디코더의 IDCT 모듈의 수행 시간과 해당 모듈을 SSE 명령어 셋을 통해 최적화 한 후의 수행 시간을 비교한 것이다. 수행 시간은 IDCT 의 변환 커널의 크기에 따라 측정되었으며, 실험 영상으로는 “BasketballDrill”이 사용되었다. 표 1 을 참조하면, HEVC 의 IDCT 커널을 SSE 를 적용하여 최적화 한 결과 기존의 HM-12.1 복호화기 IDCT 수행시간 대비 평균 3.84 배 속도가 향상됨을 알 수 있다. 그러나 4×4 커널의 경우 하나의 명령어를 통해 동시에 처리되는 데이터의 수가 많지 않아 명령어 레벨의 병렬 성능이 상대적으로 낮음을 알 수 있다. 특히, 256 비트의 레지스터가 있는 시스템에서 4×4 크기의 커널에 대해 역변환을 수행하는 경우 256 비트 레지스터의 폭을 모두 사용하지 않는다. 이러한 이유로 기존의 방식대로 구현하는 경우 256 비트 레지스터를 통한 성능 향상을 얻을 수 없다는 문제가 있다.

표 1. HEVC IDCT 커널에 SSE 를 적용한 실험결과

커널 크기	Decoding time (sec)		Speed-up
	HM-12.1	SSE	
4×4	0.46	0.13	3.45
8×8	0.51	0.09	5.53
16×16	0.30	0.07	4.14
32×32	0.13	0.06	2.25
Average	0.35	0.09	3.84

3. 제안하는 AVX2 를 이용한 IDCT 고속화

4×4 크기의 커널에 대해 순차적으로 SSE 명령어를 통해 역변환을 수행하는 기존의 최적화 방법은 256 비트 레지스터의 폭을 최대한으로 사용하지는 못한다. 본 논문에서는 이러한 문제를 해결하기 위하여 인접하는 4 개의 4×4 크기의 역양자화 된 계수 블록들을 동시에 256 비트 레지스터에 로드 한 후 동시에 4 개의 블록을 역변환하는 방법을 제안한다.

그림 1 의 (a)는 SSE 를 사용하여 블록 단위로 순차적으로 IDCT 를 수행하는 과정이다. SSE 를 사용하여 IDCT 를 수행하는 경우에는 1 개의 역양자화 된 계수 블록을 2 개의 128 비트 레지스터에 로드 한다. 그리고 4×4 커널의 전치 행렬을 128 비트 레지스터에 로드 한다. 이렇게 로드한 역양자화 계수 블록의 레지스터와 IDCT 커널의 전치 행렬을 로드한 레지스터에 대하여 1D IDCT 를 수행한다. 1D IDCT 를 수행한 결과값이 저장된 레지스터에 대해 2D IDCT 를 수행한다. 이렇게 역변환 과정이 끝난 2 개의 레지스터를 메모리에 저장한다. 그림 1 의 (b)는 AVX2 (Advanced Vector Extensions 2)를 이용하여 4 개의 4×4 블록에 대하여 IDCT 를 수행하는 과정이다. AVX2 를 이용하여 IDCT 를 수행하는 경우엔 4 개의 4×4 역양자화 계수 블록을 4 개의 256 비트 레지스터에 로드 한다. 그 후 IDCT 4×4 커널의 전치 행렬을 256 비트 레지스터에 로드 한다. 이렇게 로드한 역양자화 블록의 계수 레지스터와 커널의 전치행렬 레지스터에 대하여 1D IDCT 를 수행한다. 1D IDCT 수행한 결과 레지스터에 대해 2D IDCT 를 수행한다. 4 개의 4×4 역양자화 계수 블록에 대해 역변환 과정을 수행한 4 개의 레지스터를 메모리에 저장한다.

그림 1 의 (a)와 (b)를 비교해 보면 (a)는 4 개의 4×4 역양자화 계수 블록에 대한 IDCT 를 수행 하기 위해서는 그림 1 의 (a)과정을 4 번 반복 해야 한다. 그에 비해 AVX2 를 이용하여 4 개의 4×4 역양자화 블록에 대하여 IDCT 를 수행하는 경우 (b)의 과정 한번만 수행하면 된다. (a)의 경우 4 번 반복 수행하면 4×4 역양자화 블록을 총 8 개의 레지스터에 로드 해야 하며 매번 4×4 IDCT 커널의 전치행렬을 로드 해야 하는 단점이 있다. 반면 AVX2 를 사용하여 4 개의 4×4 역양자화 블록에 대하여 IDCT 를 수행하는 경우 4 개의 4×4 역양자화 블록을 로드 할 때 (a)의 경우보다 더 적은 레지스터를 사용한다. 또한 반복 사용되는 4×4 IDCT 커널의 전치행렬도 한번만 로드 하여 사용이 가능하다. 이러한 이유 때문에 AVX2 를 사용하여 IDCT 를 수행하는 경우에는 SSE 를 사용하여 IDCT 를 수행하는 경우보다 성능 향상을 얻을 수 있다.

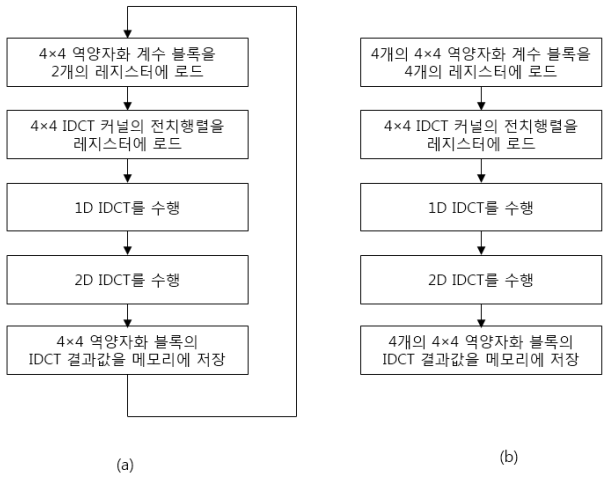

그림 1. (a) SSE 를 이용하여 IDCT 를 순차적으로 수행하는 순서도 (b) AVX2 를 이용하여 IDCT 를 한번에 수행하는 순서도

4. 실험결과 및 토의

본 논문에서 제안하는 AVX2 명령어를 이용한 IDCT 고속화에 사용된 실험 환경은 다음과 같다. Intel Core i7-4770K, 32GB RAM, MS Window 7 (64 비트) OS (Operating System) 환경에서 실험을 수행하였으며, 실험 영상으로는 HEVC common test sequence 중 Class B 와 Class C 영상을 사용하였다 [6]. 제안하는 방식의 4×4 IDCT 수행 시간은 수식 (1)을 사용하여 속도를 비교하였다.

표 2. 기존 연구와 AVX2 를 사용하여 4×4 커널에 대하여 IDCT 를 수행한 시간 비교

Class	Sequences	Decoding time (sec)			Speed-up	
		HM	SSE	AVX	HM vs. SSE	HM vs. AVX
B	Cactus	1.08	0.32	0.10	3.20	10.08
	Kimono	0.08	0.03	0.01	2.67	8.00
C	BasketballDrill	0.45	0.13	0.04	3.46	11.25
	BQMall	0.38	0.11	0.04	3.45	9.50
	PartyScene	0.64	0.18	0.06	3.56	10.66
	RaceHorses	0.15	0.04	0.02	3.75	7.50
Average					3.35	9.50

표 2 에는 HM-12.1 복호화기에서 IDCT 를 수행한 속도와 SSE 를 적용하여 IDCT 를 수행한 속도, 그리고 제안하는 방법으로 IDCT 를 수행한 시간을 초(sec) 단위로 나타내었다. 실험 결과, HEVC 디코더의 4×4 IDCT 에 SIMD 명령어 셋을 적용한 경우 기존의 HM-12.1 에 비해 평균 3.35 배 수행 속도를 향상 시킨 반면, 제안하는 방법은 HM-12.1 에 비해 평균 9.50 배 수행 속도를 향상시켰다.

$$Speed-up = \frac{Time_{reference}}{Time_{test}} \quad (1)$$

5. 결론

본 논문에서는 HEVC IDCT 의 복호화 시간을 감소시키기 위해 AVX2 명령어를 이용한 IDCT 고속화 방법에 대하여 제안하였다. AVX2 명령어를 이용하여 IDCT 를 수행한 경우 기존 HM-12.1 대비 평균 9.50 배, SIMD 명령어 셋을 사용한 경우 대비 평균 3.35 배의 속도가 향상되었다. 본 논문에서는 4 개의 4×4 블록을 동시에 IDCT 하는 방법에 대하여 제안하였으나, IDST 및 다양한 블록의 IDCT 또한 AVX2 명령어를 사용하여 동시 수행이 가능하다. 추후에는 4×4 IDST 와 8×8~32×32 블록의 IDCT 에도 AVX2 명령어를 사용하는 연구를 진행 할 계획이다.

감사의 글

본 연구는 미래창조과학부 및 정보통신산업진흥원의 대학 IT 연구센터 육성지원 사업의 연구결과로 수행되었음 (NIPA-2014-H0301-14-1018)

참고문헌

[1] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, "High Efficiency Video Coding (HEVC) text specification draft 9," document JCTVC-K1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Oct. 2012

[2] G.J Sullivan, J.-R. Ohm, W.-J Han, T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649-1668, 2012.

[3] 심동규, 조현호, "HEVC 표준 기술의 이해," 홍릉과학출판사, 2014.

[4] Leju Yan, Yizhou Duan, Jun Sun, Zongming Guo, "Implementation of HEVC decoder on X86 processor with SIMD optimization," *Institute of Computer Science and Technology* , Beijing China Oct, 2010.

[5] A. Fuldseth, G. Bjøntegaard, M. Budagavi, and V. Sze, "CE10: Core transform design for HEVC," document JCTVC-G495, Geneva, CH, Nov. 2011.

[6] Bossen, "Common test conditions and software reference configurations," document JCTVC-L1100, Geneva, CH, Jan. 2013.