멀티쓰레드와 SIMD 명령어를 이용한 실시간 H.264/AVC High 4:4:4 Predictive 디코더의 구현

Real-time H.264/AVC High 4:4:4 Predictive Decoder Using Multi-Thread and SIMD Instructions

김용환*, 김제우, 최병호, 이석필, 백준기
(Yong-Hwan Kim, Je-Woo Kim, Byeong-Ho Choi, Seok-Pil Lee, and Joonki Paik)

Abstract : This paper presents an real-time implementation of H.264/AVC High 4:4:4 Predictive profile decoder using general-purpose processors by exploiting multi-threading technique and Single Instruction Multiple Data (SIMD) instructions without any quality degradation. We analyze differences between the existing High profile and High 4:4:4 Predictive profile decoder, and show various optimization techniques to decode high fidelity and high definition (HD) video in real-time. Simulation results show that the proposed decoder can play high fidelity HD video at average 40 frames per seconds (fps) for the IBBrBP bistream and about 50 fps for the Intra-only bitstream.

Keywords: H.264/AVC, High 4:4:4 Predictive profile, Real-time decoding

I. 서론

최근에 새롭게 개정된 H.264/AVC High 4:4:4 Predictive 프로파일은 기존 2005 년 스펙의 Residual Color Transform (RCT) 기술을 삭제하고, 새로운 COMMON 코딩 모드와 INDEPENDENT 코딩 모드를 추가함으로써 이미 삭제된 H.264 High 4:4:4 profile 보다 연산량은 줄어들면서 코딩 효율은 높아졌다[1, 2]. H.264/AVC High 4:4:4 Predictive profile 의 COMMON 코딩 모드는 기존의 Y 영상 코딩 알고리듬 및 syntax 를 Cb/Cr 코딩에 그대로 이용함으로써 Cb/Cr 도 Y 의 intra 모드 및 motion vector 를 변형 없이 그대로 이용하는 방법이다. INDEPENDENT 코딩 모드는 Y/Cb/Cr planes 을 기존의 4:0:0 monochrome 코딩 방법으로 독립적인 plane 으로 코딩하는 방법이고, 인코더와 디코더가 각각의 Y/Cb/Cr planes 을 병렬 처리할 수 있다는 장점을 가지고 있다. High 4:4:4 Predictive profile 은 초고화질 응용을 위한 4:4:4 색차 포맷 및 14-bit 까지의 픽셀 깊이를 지원하기 때문에 디지털 시네마, 방송용 비디오 촬영/편집, 의료 영상, 군사용 비디오, 디지털 영상보관 등, 기존의 Baseline/Main/High profile (4:2:0 색차 포맷 및 8-bit 까지의 픽셀 깊이를 지원) 이 지원하지 못하는 모든 초고화질 영상 분야에서 사용될 수 있다. 또한 향후에는 HD 영상 이상의 고화질을 추구하는 소비자 가전 영상 분야에서도 사용될 수 있을 것이다.

요즘에 쓰이는 대부분의 범용 마이크로프로세서는 멀티미디어 응용 프로그램의 빠른 실행을 돕기 위한 멀티미디어 명령어인 SIMD (Single Instruction Multiple Data) 명령어들을 포함하고 있다. 예를 들면 인텔의 Core2 Duo/Quad 및 AMD 의 애슬론 64-X2 는 SIMD 모델인 MMX, SSE, SSE2, 및 SSE3 명령어들을 모두 포함하고 있다. SIMD 명령어들은 하나의 명령으로 동시에 여러 데이타들을 처리할 수 있기 때문에 데이터의 병렬 연산 및 메모리의 병렬 접근이 가능해서, 많은 양의 데이터를 한꺼번에 처리해야 하는 멀티미디어 응용 프로그램에서는 필수적인 요소가 됐다 [3, 4, 5, 6]. 또한 하나의 CPU 에 2 개 또는 4 개의 코어가 포함되는 multi-core CPU 가 대세를 이루고 있기 때문에 여러 코어의 연산 파워를 동시에 모두 사용하는 multi-threading 이 멀티미디어 응용 프로그램에서는 필수 요소가 되고 있는 추세다[7].

본 논문은 8-bit 이상의 화소 깊이와 4:4:4 색차 포맷을 가진 HD 영상의 화질 저하 없는 실시간 재생을 수행하기 위해 SIMD 명령어 구현, CPU 캐쉬 이용, 및 멀티쓰레딩을 이용한 디코더 구조를 소개한다.

II. 디코더 최적화 기술

A. 디코더 연산 부하 분석

High 4:4:4 Predictive 디코더의 각 모듈의 연산 부하를 측정하기 위하여 소프트웨어 성능 분석 툴을 이용하였다 [8]. 연산 부하의 분포는 기존의 High 프로파일 디코더와 크게 다르지 않지만, 가장 큰 차이는 증가된 메모리 사용량 및 접근량과 증가된 엔트로피 디코딩 연산이다[9]. 예를 들면, High 프로파일로 압축된 전형적인 HD 영상은 8-bit YCbCr 4:2:0 포맷이므로 이미지 버퍼의 크기는 1920x1080x1.5 바이트이고, 대략 10 mega bits per second (Mbps)의 비트율을 갖는다. 반면 High 4:4:4 Predictive 프로파일의 경우에는 12-bit YCbCr 또는 RGB 4:4:4 포맷이므로 이미지 버퍼의 크기는 1920x1080x3x2 바이트이고, 대략 40 Mbps 이상의 비트율을 갖는다. 그러므로 High 4:4:4 Predictive 디코더는 단순히 이미지 버퍼만을 고려해도 4 배의 메모리 접근량이 필요하고, 4 배 이상의 엔트로피 디코딩 연산이 필요하다. 그러므로 최적화 포인트는 i) 메모리 접근량을 줄이기 위해서 CPU 캐쉬를 활용, ii)효율적인 메모리 접근을 위해서 SIMD 명령어를 활용, iii) 엔트로피 디코딩을 포함하는 전체적인 연산 효율을 높이기 위한 멀티쓰레드 및 SIMD 명령어를 활용하는 것이다.

* 책임저자(Corresponding Author)
논문접수 : 20xx. x. x., 채택확정 : 200x. x. xx.
김용환, 김제우, 최병호, 이석필 : 전자부품연구원 디지털미디어연구센터 (yonghwan@keti.re.kr)
백준기, 김용환, 최병호 : 중앙대학교 첨단영상대학원.

B. SIMD 명령어를 이용한 최적화

High 4:4:4 Predictive 디코더는 최대 14-bit 화소 깊이를 처리해야 하기 때문에 하나의 화소가 16-bit로 표현된다. 따라서 inverse integer discrete cosine transform (IDCT), motion compensation (MC), deblocking filter (DF) 등을 포함하는 대부분의 연산이 32-bit 연산을 필요로 한다. 그러므로 8 화소 단위의 병렬처리를 위해서는 128-bit 의 레지스터를 갖는 SSE2 이상의 명령어를 사용해야 한다. 아래 그림 1 에 기존의 Baseline/Main/High 프로파일 디코더에서 처리하던 16-bit 연산과 High 4:4:4 Predictive 디코더의 32-bit 연산을 보여준다.

```
// Clipping operation (IDCT, MC, and DF etc)
 a) 16-bit version (SSE)
  ; (16-bit operation ~)
  packuswb    mm5, mm6;      // 0~255 clipping
  movq        [edi], mm5;     // save 8 pixels

 b) 32-bit version (SSE2)
  ;// load min/max pixel values
  movd        xmm7, max_pixel_value; //ex)1023
  pshufd      xmm7, xmm7
  packssdw    xmm7, xmm7;  // ex) [1023   ...]
  pxor        xmm0, xmm0;  // [0 0 0 0 0 ...]
  ; (32-bit operation ~)
  packssdw    xmm5, xmm6
  pmaxsw      xmm5, xmm0 ; // zero clipping
  pminsw      xmm5, xmm7; // max value clipping
  movdqa      [edi], xmm5;  // save 8 pixels
```

그림. 1. Comparison of 16-bit and 32-bit SIMD operations

그림 1 에서 보이듯이 32-bit 연산 후에 10~14 bits clipping 처리는 기존의 8-bit clipping 보다 많은 CPU 사이클을 필요로 한다. 본 논문에서는 SSE2/SSE3 명령어를 이용하여 그림 1 의 clipping 처리를 포함하는 IDCT4x4(), IDCT8x8(), LumaMC(), LumaDF(), and LumaIntraPrediction() 함수 등을 최적화하였다. 이러한 함수들은 기존의 16-bit SSE 연산을 32-bit SSE2/SSE3 연산으로 변경하여 구현하였다[5, 6].

C. CPU 캐쉬를 활용한 최적화

High 4:4:4 Predictive 디코더에서 10-bit 4:4:4 포맷의 HD (1920x1080) COMMON 모드 영상 디코딩의 경우에 이미지 버퍼 이외에 필요한 부가 메모리의 크기 및 접근량은 아래 표 1 과 같다.

표 1. 10-bit 4:4:4 HD 영상 디코딩을 위한 부가 메모리

버퍼	크기[바이트]	매크로블록당 최대 접근량
Intra prediction mode	130,560 (480x272)	48
MV (L0/L1)	1,044,480	160
Reference index (L0/L1)	65,280 (240x136x2)	160
Number of non-zero coefficient (Y/Cb/Cr)	391,680	144

INDEPENDENT 모드의 경우에는 표 1 에서 "Number of non-zero coefficient" 만 제외하고, 3 배의 메모리 크기 및 접근량을 필요로 하기 때문에, 결과적으로 싱글 쓰레드에서는 COMMON 모드보다 조금 느린 디코딩 속도를 보인다. 매크로블록 디코딩 과정에서 표 1 의 모든 버퍼를 자주 접근하는 것은 계속적인 CPU 캐쉬 미스가 생김으로 인해 반복적인 메인 메모리 접근 지연이 발생할 확률이 커지게 된다. 따라서 각각의 버퍼에 해당하는 작은 캐쉬 버퍼를 미리 만들어 놓고, 매크로블록 디코딩 전에 주 버퍼로부터 데이터를 읽어와서 캐쉬 버퍼를 채우고, 디코딩 과정에서는 캐쉬 버퍼만 접근(읽기/쓰기)하고, 매크로블록 디코딩이 끝난 후에는 캐쉬 버퍼의 데이타를 주 버퍼에 써 주면 매크로블록당 단 몇 번의 메모리 접근 지연을 예상할 수 있다. 예를 들어 "intra prediction mode buffer"의 경우에 캐쉬 버퍼의 크기는 25 바이트(5x5)이면 충분할 정도로 작기 때문에 디코딩 과정에서 캐쉬 버퍼 자체가 CPU 캐쉬에 머무를 가능성은 매우 커지게 되기 때문에 CPU 캐쉬 미스 발생 확률을 크게 줄일 수 있게 된다. 또한 캐쉬 버퍼를 이용함으로써 CPU 연산의 큰 부하 중인 하나인 조건 분기를 크게 줄일 수 있는 장점도 갖게 된다. 즉, 표 1 의 버퍼들을 접근할 때 항상 픽쳐 경계, 슬라이스 경계, 블록 유효성 등 많은 조건들을 체크해야 하지만, 매크로블록 디코딩 전에 캐쉬 버퍼를 채울 때 조건 검사를 한 번에 모아서 할 수 있기 때문에, 실제 매크로블록 디코딩시에 분기 예측 오류(branch mispredition)을 크게 줄일 수 있다. 순수하게 이러한 캐쉬 버퍼 기술에 의해서만 디코더의 속도가 대략 2 배 정도 빨라지게 된다. 한편 CABAC 스트림 디코딩의 경우에는 motion vector difference (MVD) 와 Direct8x8 mode 버퍼의 경우에 캐쉬 버퍼를 이용할 수 있다.

D. 멀티쓰레드를 이용한 최적화

High 4:4:4 Predictive 프로파일의 INDEPENDENT 코딩 모드는 Y/Cb/Cr 또는 R/G/B 각각의 plane을 독립적으로 코딩하기 때문에 멀티쓰레드에 적합한 구조를 가지고 있다. 아래 그림 2 에 4 개의 쓰레드를 이용한 멀티쓰레드 디코더 구조를 보인다. 그림에서 Thread1 은 파일 읽기, NALU 헤더 분석, 슬라이스 헤더 분석, 및 RBSP 변환을 수행한 후에 NALU 비트 스트림 버퍼에 각 plane 별로 NALU를 저장한다. 특히 디코딩을 시작하기 전에 SPS/PPS를 각각의 plane별 NALU 버퍼에 복사해 줌으로써, 각각의 4:0:0 디코더가 Thread2, Thread3, Thread4 에서 완전하게 독립적으로 실행될 수 있다. 여기에서 한 가지 고려사항은 재생단에서 막대한 메모리 복사를 피하기 위해서 프레임별로 쓰레드 동기를 맞추도록 했다. 즉, 하나의 R/G/B plane 디코딩이 모두 끝난 후에 프레임을 구성해서 재생한 후에 다시 다음 plane들을 디코딩하게 된다.

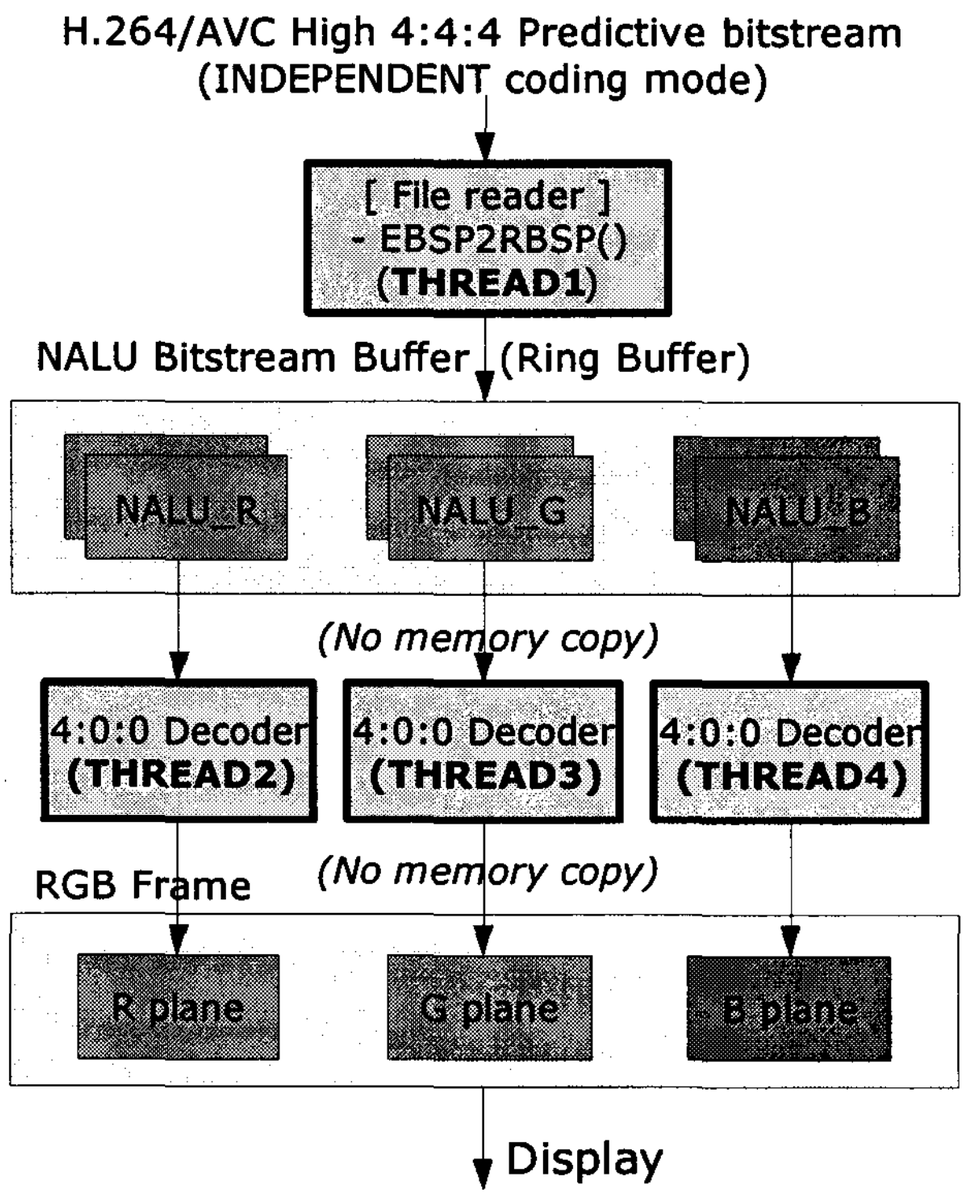

그림 2. 멀티쓰레드를 이용한 High 4:4:4 Predictive 디코더의 구조

III. 실험 결과

우리는 제안된 디코더를 범용 워크스테이션[1]에서 구현했다. 사용된 실험 영상과 인코딩 파라미터는 표 2 에 보인다[10]. 참고로 우리는 최신 표준에 맞추기 위해서 참조 소프트웨어의 여러 버그를 수정하였다 [1, 11].

표 2. 테스트 영상 및 인코딩 파라미터

[Sequences: 10-bit RGB, 1920x1080, 24 fps, 57 frames] 1) Viper camera: Freeway (FW), Plane (PL), Wave (WV) 2) Film scan: Man in restaurant (MIR; 1920x896), Playing cards (PC), Rolling Tomatoes (RT)	
Profiles	CAVLC 4:4:4 Intra / High 4:4:4 Predictive
Coding structure	Intra-only (5 slices/picture) / IBBrBP
Entropy coding	CAVLC
Deblocking filter	Off (Intra-only), On (IBBrBP)
QP	18, 24, 30 (+2 for the B slice)
Intra period	3 (IBBrBP) (0.5 seconds period)
Num of reference	3
Direct mode	Spatial

우리는 COMMON 모드와 INDEPENDENT 모드를 싱글 쓰레드로 구현하였고, INDEPENDENT 모드는 멀티쓰레드 방식으로도 구현하였다. 디코딩 속도 테스트는 4 번의 실행 중에, 첫 번째 실행을 제외한 3 번의 실행 결과에서 중간 값을 선택하였다. 디코더 실행 중에는 화면 재생과 파일 저장을 하지 않았고, 시간은 Windows 환경에서 가장 정확하다고 알려져 있는 QueryPerformanceCounter() 함수를 써서 측정하였다.

[1] Com piled by Visual C++ 8.0 SP1 on a Windows XP SP2 workstation with two Zeon 5160 CPU and 4GB DDR2 667MHz RAM

아래 표 3 과 4 에 디코더 속도 실험 결과를 보인다.

표 3. IBBrBP 스트림의 디코더 속도

Seq.	QP	PSNR	CB	STC	STI	MTI
FW	18	44.8	89.5	15.1	12.9	**34.5**
	24	40.1	34.9	21.9	18.6	**46.6**
	30	36.4	14.7	28.5	23.3	**55.3**
PL	18	45.3	58.1	13.0	12.0	**30.8**
	24	41.7	16.8	17.3	15.6	**38.5**
	30	39.0	6.1	19.5	17.1	**42.8**
WV	18	45.4	49.3	14.7	13.5	**34.1**
	24	42.2	13.0	21.3	18.8	**46.5**
	30	39.9	4.9	26.3	23.5	**55.2**
MIR	18	43.7	54.8	18.9	15.4	**34.3**
	24	40.4	11.3	33.5	26.5	**58.7**
	30	38.1	4.0	43.4	34.0	**77.7**
PC	18	44.0	164.7	10.3	9.2	**24.1**
	24	39.5	74.7	13.4	12.1	**29.7**
	30	35.6	26.6	19.0	19.0	**32.5**
RT	18	43.4	99.3	12.8	11.1	**30.7**
	24	39.6	22.9	21.3	16.5	**34.0**
	30	38.2	5.7	30.7	34.0	**55.5**

-PSNR: average PSNR of R, G, and B signal [dB]
-CB: COMMON mode bitrates (INDEPENDENT mode bitrates are similar) [Mbps]
-STC: Single-threaded COMMON mode decoding speed [fps]
-STI: Single-threaded INDEPENDENT mode decoding speed [fps]
-MTI: Multi-threaded INDEPENDENT mode decoding speed [fps]

표 4. Intra-only 스트림의 디코더 속도

Seq.	QP	PSNR	CB	STC	STI	MTI
FW	18	46.6	217.2	14.5	14.3	**35.7**
	24	41.7	130.2	19.5	19.0	**40.8**
	30	37.5	76.2	25.3	24.5	**48.4**
PL	18	47.1	120.6	19.8	19.4	**43.7**
	24	43.1	64.5	27.0	26.4	**50.5**
	30	39.7	36.1	34.2	33.8	**52.5**
WV	18	47.1	91.0	22.8	22.8	**48.0**
	24	43.6	41.8	33.4	33.2	**57.4**
	30	40.8	20.7	42.8	43.7	**73.7**
MIR	18	45.6	130.3	20.9	20.2	**50.1**
	24	41.6	64.8	30.6	29.4	**58.2**
	30	38.5	35.3	40.5	39.0	**71.2**
PC	18	45.5	211.0	14.8	14.6	**31.8**
	24	40.9	113.9	21.3	21.2	**44.0**
	30	37.5	61.0	28.7	28.7	**54.2**
RT	18	45.0	129.5	18.9	19.1	**45.3**
	24	41.0	42.0	31.0	32.0	**56.6**
	30	39.1	18.2	42.8	48.5	**75.7**

표 3 과 4 의 실험 결과로부터 알 수 있듯이, 제안된 디코더는 평균 40Mbps 의 비트율을 갖는 초고화질 HD 스트림을 IBBrBP 구조에서는 평균 40 fps, Intra-only 구조에서는 평균 50 fps 로 재생할 수 있다. 위의 표들에서는 INDEPENDENT 모드 스트림의 비트율은 COMMON 모드와 거의 동일하기 때문에 따로 표시하지는 않았다.

IV. 결론

본 논문에서는 H.264/AVC High 4:4:4 Predictive 디코더의 최적화 포인트를 분석하고, 실시간 디코더를 구현하였다. SSE2/SSE3 명령어를 광범위하게 이용하고 멀

티쓰레드 기술을 적용하여, 범용 프로세서에서 초고화질 HD 영상을 실시간 재생이 가능하도록 최적화하였다. 제안된 디코더는 디지털시네마, 의료 영상, 군사용 영상 분야에 바로 사용할 수 있다. 현재 CABAC 디코딩은 CAVLC 디코딩보다 20~30% 정도 느리지만, 향후 출시될 SSE4 명령어를 가진 범용 CPU 에서는 CABAC 스트림도 실시간 디코딩이 충분히 가능할 것으로 예상된다.

참고문헌

[1] Joint Draft 6 of "New profiles for professional applications" amendment to ITU-T Rec. H.264 & ISO/IEC 14496-10 (Amendment 2 to 2005 edition), JVT-V204, Marrakech, Jan. 2007.

[2] ITU-T Recommendation H.264, "Advanced video coding for generic audiovisual services," ITU-T, March 2005.

[3] Richard Gerber, "The Software Optimization Cookbook," Intel Press, 2002

[4] Intel Corp., "Intel 64 and IA-32 Architectures Optimization Reference Manual," Order Number: 248966-015, May.2007

[5] X. Zhou, et al., "Implementation of H.264 Decoder on General-Purpose Processors with Media Instructions," *Proc. SPIE Conference on Image and Video Communication and Processing,* vol. 5022, Jan. 2003

[6] Y. H. Kim, J. W. Yoo, S. W. Lee, J. Paik and B. Choi, "Optimization of H.264 Encoder Using Adaptive Mode Decision and SIMD Instructions," *Proc. IEEE ICCE,* Jan. 2005

[7] S. ge, X, Tian, and Y.K. Chen, "Efficient Multithreading Implementation of H.264 Encoder on Intel Hyper-Threading Architectures," *Proc. ICICS-PCM,* Dec. 2003

[8] Intel Corp., Intel VTune Performance Analyzer, version 9.0, 2007.

[9] T. Bhatia, "Optimization of H.264 High Profile Decoder for Pentium 4 Processor," Master thesis of Electrical Engineering, The University of Texas at Arlington, Dec. 2005.

[10] JVT, "Ad-Hoc Group Report: Study of 4:4:4 functionality," JVT-R009, Bangkok, Jan. 2006.

[11] JVT, "Joint 4:4:4 Video Model (JFVM) 6 Reference Software," JVT-V206, Marrakech, Jan. 2007.

김 용 환

1996 년 중앙대학교 전기공학과 졸업(학사), 1998 년 중앙대학교 전기공학과 졸업(석사), 1999 년~2001 년 성진씨앤씨 대리, 2001 년~현재 전자부품연구원 디지털미디어연구센터 선임연구원, 2004 년~현재 중앙대학교 첨단영상대학원 박사과정. 주 관심분야는 2D/3D 영상 압축 알고리듬 및 비디오 코덱 최적화 구현

최 병 호

1991 년 한양대학교 전자공학과(공학사). 1993 년 한양대학교 전자공학과(공학석사). 2001 년 KAIST 전산학 박사중퇴. 2006 년 중앙대학교 첨단영상대학원(공학박사수료). 1993 년~1997 LG 전자 비디오 연구소 주임연구원. 1997 년 ~ 현재 전자부품연구원(KETI) 디지털미디어연구센터 책임연구원. 주 관심분야는 영상압축, 3D 비디오 처리, 영상 ASIC

백 준 기

1984 년 서울대학교 제어계측공학과 졸업(학사), 1987 년 노스웨스턴대학교 전기 및 컴퓨터 공학과 졸업(석사), 1990 년 노스웨스턴대 전기 및 컴퓨터공학과 졸업(박사), 2007 년 현재 중앙대학교 첨단영상대학원 영상공학과 교수. 주 관심분야는 영상복원, 신호처리, 반도체 2004 년~현재: MPEG Forum 운영위원 2002 년~현재: TTA 지원 국제표준전문가

김 제 우

1999 년 서울시립대학교 제어계측공학과 석사 졸업. 1999 년 ~ 현재 전자부품연구원 디지털미디어연구센터 책임연구원. 주 관심분야는 비디오 코덱, SVC, 멀티미디어 전송.

이 석 필

1990 년 연세대학교 전기공학과 졸업(학사), 1992 년 연세대학교 대학원 전기공학과 졸업(공학석사), 1997 년 연세대학교 대학원 전기공학과 졸업(공학 박사)
1997 년~2002: 대우전자 영상연구소 선임연구원
2002 년~현재: 전자부품연구원 디지털미디어연구센터 센터장
2000 년~현재: 차세대방송표준포럼 TV Anytime 분과위원회 위원장
2004 년~현재: MPEG Forum 운영위원
2002 년~현재: TTA 지원 국제표준전문가
2004 년~현재: TTA 디지털 TV 프로젝트 그룹 부의장
주 관심분야: 개인 맞춤형 방송 솔루션, 데이터 방송, 멀티미디어 통신, 인공지능