

최소의 화질 열화와 함께 실시간 동작이 보장되는 H.264 동영상 복호기

정희원 김 종 찬*, 김 두 리**, 이 동 호***

An H.264 Video Decoder which Guarantees Real-Time Operation with Minimum Degradation

Jong-chan Kim*, Du-ri Kim**, Dong-ho Lee*** *Regular Members*

요 약

H.264 기술은 차세대 동영상 코덱 표준의 핵심으로 간주되고 있다. 유럽을 포함하여 많은 나라에서는 HD 방송을 위한 동영상 코덱 표준으로 H.264 기술을 사실상 지정해 놓고 있는 실정이다. 하지만 복잡한 알고리즘 사용으로 인해 HD급 영상의 경우에는 아직도 데스크탑 컴퓨터에서조차 실시간 복호화가 어려운 상황이다. 본 논문에서는 실시간으로 동작이 보장되는 H.264 소프트웨어 동영상 복호기를 구현하기 위해서 복호화 과정의 일부를 제한하고, 이에 따른 화질열화가 최소가 되는 알고리즘들을 적응적으로 선택하는 H.264 복호기를 제안한다. 제안하는 H.264 복호기는 PC 환경에서 모의실험을 통해 성능을 비교 및 검증하였다. 그 결과 실시간 복호화가 어려운 환경에서 제안하는 복호기를 사용하였을 경우 대부분 최소한의 화질 열화와 함께 실시간 복호화를 만족하는 결과를 보였다.

Key Words : H.264, FF-MPEG, HD video, decoding method

ABSTRACT

H.264 technology is considered as the heart of the next-generation video codec standard. Europe and other countries have actually specified H.264 technology as the video codec standard for HD broadcasting. However, due to the complexity of algorithm, it is still a difficult job to implement HD-level H.264 decoders in real-time software. In this paper, I have restricted a part of the decoding process, in order to implement an H.264 software video decoder which guarantees a real-time operation, and suggest an H.264 decoder that adaptively selects the algorithm to minimize image degradation. Performance of the suggested H.264 decoder was compared and verified through a PC simulation. As a consequence, when the suggested decoder was used in an environment where real-time decoding was difficult, it has achieved the minimal image degradation as well as real-time decoding in most cases.

I. 서 론

동영상 압축에 대한 표준은 1980년대부터 ISO와 ITU에 의해 꾸준히 진행되어 왔으며 MPEG-1에서

부터 H.264(MPEG-4 Part10/AVC)에 이르기까지 다양한 압축 표준을 선보이고 있다. 우수한 압축 성능으로 인해 H.264는 모바일 서비스와 같이 높은 데이터 압축률을 요구하는 환경에서 최적의 서비스를

* LG전자 DTV연구소 (hotkjh79@lge.com), ** 삼성전자 VD 사업부 (duri.kim@samsung.com)

*** 한양대학교 전자컴퓨터공학부 (dhlee@image.hanyang.ac.kr)

논문번호 : KICS2008-05-228, 접수일자 : 2008년 5월 21일, 최종논문접수일자 : 2008년 9월 8일

지원할 수 있다. 현재 휴대폰으로 볼 수 있는 대부분의 동영상 콘텐츠는 MPEG-4 기술이 적용되어 있지만 H.264 기술을 적용할 경우 현재보다 약 2배의 화질 또는 속도로 서비스가 가능하다⁷⁾.

H.264 기술은 기존 다른 코덱들과 동등한 화질을 구현함에 있어 50%이상의 높은 압축률을 보이고 있다. 하지만 기존 코덱인 MPEG-2와 비교하였을 때 구현과 관련된 전체적인 복잡도가 4배 이상 증가하는 단점이 있다. 이는 고효율의 압축 성능을 위해서 기존의 동영상 압축 규격과는 다른 복잡한 새로운 기법을 적용함으로써 인해서 생기는 문제점이다³⁾⁴⁾.

기존에는 동영상 복호기를 하드웨어적으로 구현을 하는 것이 보편적이었으나 최근에는 반도체의 발달로 인해 프로세서의 성능이 크게 발전함과 동시에 호환성과 확장성 등을 위해 이를 소프트웨어적으로 구현을 하는 추세에 있다⁵⁾. 하지만 H.264의 복잡한 알고리즘 사용으로 인해, HD급 영상의 경우에는 아직도 데스크탑 컴퓨터에서조차 실시간 복호화가 어려운 실정이다. 특히, PMP(Portable Multimedia Player)와 같은 실시간 복호화가 요구되지만 제한된 성능의 프로세서를 지원하는 휴대용 멀티미디어 단말기에서는 화질의 열화를 최소화 시키면서 실시간 복호화가 가능한 복호기의 개발이 요구된다.

본 논문에서는 제한된 성능의 프로세서 환경에서 동영상 복호기를 실시간으로 구현할 수 있도록 복호화 과정의 일부를 제한하고, 이에 따른 화질 열화가 최소가 되는 알고리즘을 포함시킨 H.264 복호화 방법을 제안한다. 본 논문에서 이용한 소프트웨어 소스 코드는 FF-MPEG의 H.264 복호기를 최적화한 소스이며, baseline 프로파일을 대상으로 하고 있다. 먼저 모의실험을 통하여 화질을 최소화 하면서 연산을 줄일 수 있는 간이형 복호화 방법과 이들의 조합으로 구성되는 복호화 방법들을 결정하였다. 다음으로 프로세서의 복호화 성능을 측정하여 이에 최적화된 간이형 복호화 방법을 선택하여 복호화를 수행하는 적응적인 H.264 복호기를 제안하였다.

II장에서는 대상 복호기의 구조와 복잡도를 설명하고 복호화 과정의 일부를 제한하게 된 모듈의 선택 이유를 제시한다. III장에서는 H.264 동영상 복호기를 제한된 환경에서 실시간으로 구현할 수 있도록 복호화 과정의 일부를 제한하는 방법들을 제시하고, 이 방법들을 조합하여 이에 따른 화질 열화가 최소가 되는 알고리즘을 제안한다. IV장에서는 제시한 알고리즘을 하나의 복호기로 만들어 복호화에 적응적으로 적용시킬 수 있는 효과적인 방법에 대하여 제

시하고, V장에서 모의실험을 통한 분석을 한다. 마지막으로 VI장에서 결론을 맺는다.

II. H.264/AVC 동영상 복호기

그림 1에서와 같이 H.264 복호기는 압축된 비트스트림을 받아 데이터 요소들에 대한 가변길이 복호화를 수행하여 양자화된 계수 X를 생성한다. 생성된 계수들은 역양자화(Q-1)되고 역변환(T-1)되어 오차블록 D'n이 생성된다. 복호기는 비트스트림으로부터 복호화된 헤더 정보를 사용하여 인코더에서 생성된 것과 동일한 예측 블록(P)를 생성한다. P는 D'n에 더해져서 uF'n를 생성하며, uF'n는 필터를 거쳐 각각의 복호화된 블록 F'n를 생성한다⁶⁾.

본 논문에서는 리눅스 기반의 공개 소스인 FF-MPEG 통합 코덱을 분석하여 H.264 복호기만을 분리하여 사용하였다. FF-MPEG은 소프트웨어 전문가들이 지속적인 업데이트를 통해 최적화가 이루어지고 있으며 성능 또한 기존 공개 소스들 중에서 가장 성능이 앞서고 있다. 그림 2에서 보듯이 전체 H.264 동영상 복호기에서 디블록킹 필터의 복잡도가 35%, 움직임 보상에 29%, 엔트로피 코딩에서 17%, 역양자화 및 역변환 과정에서 6%정도의 복잡도를 차지하고 있다.

본 논문에서는 역변환, 움직임 보상, 그리고 디블록킹 필터 부분의 복호화 과정을 제한하는 알고리즘

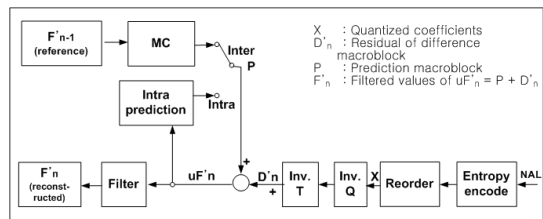


그림 1. H.264 복호기의 블록도

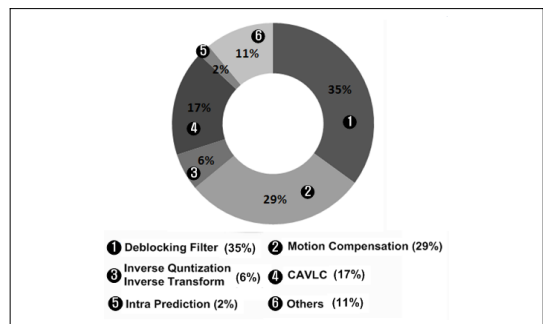


그림 2. H.264 복호기의 각 모듈별 복잡도

들을 제안 하였다. 특히, 움직임 보상과 디블록킹 필터 부분은 전체 복호화과정의 70% 가량을 차지하고 있으며 상대적으로 화질 열화가 적으면서 연산량을 대폭 줄일 수 있는 간이형 알고리즘들이 존재한다.

Ⅲ. 간이형 복호화 알고리즘

3.1 모듈 별 간이형 복호화 적용방법

화질을 최소화하면서 연산량을 줄이는 간이형 복호화 알고리즘 들을 유도하기 위한 첫 번째 단계로 H.264 복호기를 구성하는 단위 별로 화질의 손실을 감수하면서 연산량을 줄일 수 있는 가능한 복호화 방법 들을 정리하면 다음과 같다.

3.1.1 역변환 과정

DC 계수 값은 블록의 평균 밝기를 나타내기 때문에 AC 계수보다 중요하다 할 수 있다. Intra 블록의 경우, 예측과정에 있어 큰 영향을 미치게 되므로 이 부분의 작은 화질 열화는 전체 복호기에 큰 영향을 미치게 된다. 또한 영상의 색차 성분의 경우 휘도 성분에 비해 사람 눈에 미치는 영향이 적다.

역변환 과정의 간이형 복호화 방법은 화질열화에 영향을 덜 미치는 Inter블록과 색차 성분의 경우에 대해 일부 AC 계수를 제한하여 역변환 하는 방법을 적용하였다.

3.1.2 휘도성분의 움직임보상 과정

H.264의 움직임 보상 과정에서는 예측보간 신호의 생성에 많은 연산량이 소요된다. 휘도 성분의 경우 1/4 화소 단위의 보상을 적용한다. 첫 번째 단계로 정수화소 신호로부터 6탭 FIR필터를 사용하여 1/2화소 예측신호를 생성하고 그 다음 과정으로 2탭의 평균치 필터에 의해 1/4화소 예측신호를 생성한다.

이 과정에서의 연산량을 줄이기 위해 6탭 필터를 대신하는 4가지 간이형 복호화 방법을 제안한다. 첫째, bicubic interpolation을 적용하여 4탭 필터를 적용한다. 둘째, bilinear interpolation을 적용하여 2탭 필터를 사용한다. 셋째, 6탭 필터를 사용하지 않고 블록의 일부를 정수화소로 대입하여 사용한다. 그 결과 해당 정수 화소의 주변 3x3 블록이 이 값으로 대체되게 된다. 예측블록 내의 나머지 값들은 기존의 방식을 사용해 보간 된다. 넷째, 블록전체를 정수 화소로 패딩 시키는 방법을 적용한다.

3.1.3 색차성분의 움직임보상과정

색차 성분의 예측보간 신호는 정수화소를 1/8화소

로 선형 보간하여 생성한다. 간이형 복호화 방법에서는 1/8화소 보간을 대신해 평균값을 사용하는 2탭 필터와 계산 없이 정수화소를 복사하여 사용하는 방법을 적용한다.

3.1.4 디블록킹 필터 과정

H.264에서는 블록 간의 왜곡현상을 감소시키기 위해 각각의 복호화된 매크로블록에 필터가 적용된다. 이를 디블록킹 필터라 하는데, 이 부분에서는 BS (Boundary Strength)값에 따라 필터 적용의 강도가 달라지게 된다. BS 값이 높을수록 왜곡 현상이 심한 경우를 나타내며, 필터 적용의 강도가 높아진다.

간이형 복호화 방법에서는 BS 값을 이용하여 필터 적용의 강도를 조절하였다. 또한 디블록킹 필터 함수들의 내부는 각각 Intra, Inter에 따라 나뉘어진다. 이 구조를 이용해 Intra 블록의 경우만 디블록킹 필터를 적용하는 솔루션을 제안한다. 또한 디블록킹 필터는 경계 블록에서의 왜곡현상을 감소시키기 위한 부분이므로 복호화 자체에는 큰 영향을 미치지 않는다. 따라서 필터 적용자체를 제외하는 방법 또한 제안한다.

3.1.5 모듈 별 간이형 복호화 적용방법 요약

표 1에는 위에서 제안한 복호기를 구성하는 단위 별로 여러 간이형 복호화 적용방법을 요약하고 있다. 역변환 부분에서는 DC계수만 변환하는 방법을 이용하여 1-A~1-C의 3가지, 휘도성분의 움직임 보상부분에서는 6탭 필터를 대신하여 4탭, 2탭, 정수화소를 사용하는 방법을 이용하여 2-A~2-D의 4가지, 색차 성분의 움직임 보상부분에서는 1/8화소 선형보간을 대신하여 2탭, 정수화소를 사용하는 방법을 이용하여 3-A~3-B의 2가지, 디블록킹 필터부분에서는 BS 값

표 1. 모듈별 간이형 복호화 적용방법 요약

DCT	1-A	chroma에서 DC만 역변환
	1-B	inter에서 DC만 역변환
	1-C	inter, chroma에서 DC만 역변환
MC (Luma)	2-A	luma 4 tap filter 사용
	2-B	luma 2 tap filter 사용
	2-C	luma integer pel 사용
	2-D	luma integer pel copy
MC (Chroma)	3-A	chroma 2 tap filter 사용
	3-B	chroma integer pel 사용
deblocking	4-A	BS가 2 또는 1일 때 horizontal filter만 적용
	4-B	BS가 2 또는 1일 때 vertical filter만 적용
	4-C	BS가 2 또는 1인 경우를 제외
	4-D	BS가 1,2,3인 경우를 제외
	4-E	INTRA에 대해서만 적용
	4-F	디블록킹 필터를 제외

을 사용하는 방법, 인트라 블록에만 적용하는 방법 등을 이용하여 4-A~4-F의 6가지, 이렇게 전체적으로 총 15가지의 가능한 방법을 제안한다.

3.2 간이형 복호화 적용방법의 조합 및 선택

이번 절에서는 앞서 설명한 간이형 복호화 방법을 조합하여 만든 복호화 알고리즘을 제안하고 있다. 독립적으로 적용할 수 있는 복호화방법의 그룹은 (1.역변환, 2.움직임 보상-휘도성분, 3.움직임보상-색차성분, 4.디블록킹 필터) 이렇게 4개로 나뉘 볼 수 있다. 이 그룹을 묶어 2개, 3개, 4개의 조합을 만들었다. 앞서 만들었던 15개의 방법과 2개 조합(1-2, 1-3, 1-4, 2-3, 2-4, 3-4)에서의 80개, 3개 조합(1-2-3, 1-2-4, 1-3-4, 2-3-4)에서의 180개, 그리고 4개 조합(1-2-3-4)에서의 144개, 총 419개의 조합을 만들 수 있고 그 만큼의 간이형 복호화 알고리즘이 존재하게 된다. 그러한 많은 복호화 알고리즘 중에서 일정한 연산량을 줄일 때 화질 열화가 최소가 되는 알고리즘을 유도하기 위해 위와 419 가지의 복호화 알고리즘을 구현하여 SD급과 HD급의 H.264 테스트 영상을 이용하여 성능을 분석하였다. H.264 동영상 복호기의 특성 때문에 다른 영상에 대해서도 큰 차이가 없이 유사한 실험 결과를 얻을 수 있었다. 1차적으로 이들 419 개의 조합 중에서 최소의 화질을 보장하기 위해 PSNR이 30 dB 이상인 56개의 조합 만을 추출했다.

추출된 56개 중에서 간이형 복호화 알고리즘으로 사용할 조합을 선택하게 되는데, 우선 각 복호기의 속도 향상률에 따라 재배열을 시킨 뒤에 이를 기준으로 구간을 나누었다. 그 다음 구간 안에서 PSNR을 기준으로 몇 개의 조합을 선택하고, 선택한 영상의 장면 비교를 통해 주관적인 화질을 평가하였다. 이는 PSNR 값이 절대적이지 않기 때문에 눈으로 봤을 때 나온 성능을 보이는 조합을 선택하기 위함이다. 이 때 비교 장면은 가장 오차가 많이 생길 수 있는

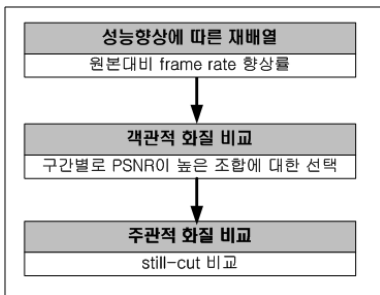


그림 3. 사용할 알고리즘 선택 과정

인트라 프레임 직전의 장면을 참고로 사용했다. 이렇게 되면 특정한 속도 향상률을 만족하는 복호화 알고리즘 중에서 화질 열화가 최소화되는 간이형 복호화 알고리즘이 결정되게 된다.

실험결과 다른 모듈에 비해 디블록킹 필터 부분을 제한할 때 생기는 속도 향상률과 PSNR이 높게 나왔다. 간이형 복호화 방법의 조합에서도 디블록킹 필터의 제한을 이용한 방법이 월등한 성능을 보였다. 표 2는 4개의 다른 H.264 동영상에 위의 과정을 적용하여 유도해낸 각기 다른 수준의 최적 간이형 복호화 알고리즘 들이다. 알고리즘1에서 알고리즘8까지 PSNR을 30.86dB에서 48.77dB로 유지하면서 원본 복호기의 복호화 속도에 비해 최대 59.28%에서 최소 6.75%까지 성능을 향상 시킬 수 있다.

그림 4와 그림 5는 위에서 제안한 간이형 복호화 알고리즘을 참조 소스에 적용하여 구현한 결과에 대해 초당 복호화되는 프레임 수를 측정하여 이를 원

표 2. 간이형 복호화 알고리즘

알고리즘	속도 향상 평균 (%)	PSNR 평균 (%)	적용된 디코딩 방법
algorithm1	6.75	48.77	4-A
algorithm2	17.42	45.17	4-C
algorithm3	22.87	46.36	4-E
algorithm4	27.22	42.61	4-D
algorithm5	37.41	41.54	4-F
algorithm6	48.46	35.20	4-F 2-A
algorithm7	52.78	32.09	4-F 2-A 3-B
algorithm8	59.28	30.86	4-F 2-B

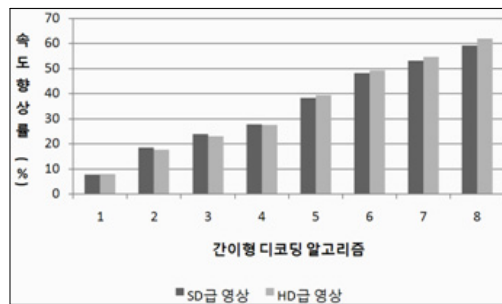


그림 4. 간이형 복호화 알고리즘 적용결과 (속도향상률)

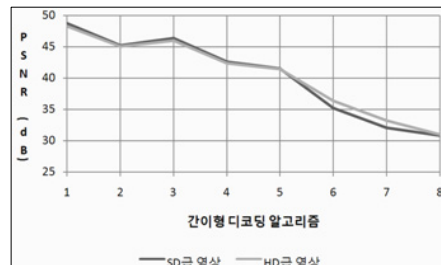


그림 5. 간이형 복호화 알고리즘 적용결과 (PSNR)

본 소스와 비교한 결과를 보여준다. 사용된 PC의 성능은 Intel core2 duo CPU 1.86GHz에 메모리는 1Gbyte이다. 테스트 비트스트림은 H.264 baseline 프로파일로 인코딩한 SD급 2개와 HD급 2개의 영상이다. 알고리즘 1에서 알고리즘 8까지 속도 향상이 높아지고 이에 따라 PSNR은 감소하는 모습을 보이고 있다.

결론적으로 위에서 설명한 8개의 복호화 알고리즘은 주어진 복호화 속도 향상 비율을 만족하면서 화질 열화를 최소화 시키는 알고리즘들이라 할 수 있다. 복호화 속도를 약 60 % 정도 향상 시키더라도 복호화된 영상의 화질은 PSNR을 기준으로 최소 30 dB 이상의 영상을 복원할 수 있게 된다.

IV. 실시간 동작이 보장되는 H.264 동영상 복호화 방법

4.1 간이형 복호화 알고리즘 들의 구현

먼저 FF-MPEG의 H.264 복호기 소스를 이용하여 각 모듈에 대해 간이형 알고리즘 들을 구현하였다. 휘도성분의 움직임보상 과정에서는 최상위 함수에서 매개변수를 이용한 배열함수의 사용을 통해 연산량의 증가 없이 알고리즘을 적용시켰다. 하지만배열함수의 생성으로 인하여 파일 사이즈는 14% 정도 증가한다. 색차성분의 움직임보상 과정에서는 조건문을 이용한 적용에 의해 약간의 연산량이 증가하였다. 마지막으로 디블록킹 필터 과정에서도 배열함수를 적용하여 이로 인한 연산량의 증가 없이 알고리즘을 적용 시켰고, 파일사이즈는 10.5% 증가하는 결과를 가져왔다.

표 3에는 이러한 간이형 복호기를 1개의 프로그램으로 통합하여 구현한 결과를 나타내었다. 전체적으로 실행파일의 사이즈는 기존에 비해 약 26.3% 증가하였고, 조건문의 개수는 한 프레임 당 1000~3000여 개 정도 증가하여 복호화 과정에 큰 부담을 주지 않는 결과를 보였다.

4.2 간이형 복호화기의 실시간 적용 방법

그림 6에는 간이형 복호화 알고리즘의 실시간 적

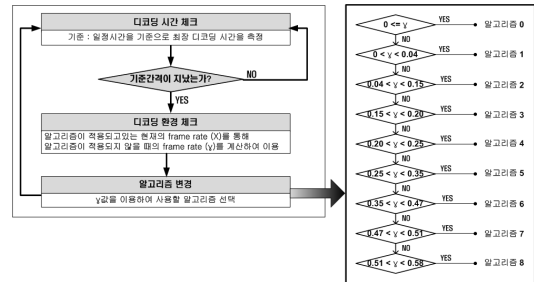


그림 6. 간이형 복호기의 실시간 적용방법

용방법의 전체적인 흐름을 나타내었다. 본 논문에서는 매 프레임마다 복호화 시간을 체크하고 일정간격 안에서의 최장 복호화 시간을 기준으로 복호기가 현재 실시간 복호화를 만족하는지 판단한 후, 적용될 알고리즘을 선택하였다.

복호화 알고리즘 변경을 위한 기준시간을 구한 뒤에는 복호기가 현재 실시간 복호화를 만족하는지 판단해야 하고 상태에 맞는 알고리즘을 선택해야 한다. 복호화 시간을 계산하는 과정에서 얻어지는 정보로는 알고리즘이 적용되고 있는 현재의 초당 프레임 수와 적용되고 있는 알고리즘의 증가비율 α , 현재 상태에서 실시간 복호화를 위해 추가로 증가 시켜야 할 비율 β 를 얻어낼 수 있다. 이 정보들을 이용하여 다음에 적용될 증가비율 $\gamma(\alpha + \beta)$ 를 얻을 수 있다.

앞에서 얻은 다음에 적용될 증가비율 γ 를 기준으로 다음에 적용될 알고리즘을 선택하게 된다. 다음에 적용될 알고리즘은 조건문을 사용하여 γ 값을 보고 바로 해당 알고리즘을 선택하게 되는 구조로 되어있다. 알고리즘의 적용 비율은 표 4와 같이 안정적인 실시간 동작을 위하여 앞에서 구한 비율보다 좀더 포함 범위를 넓힐 수 있는 기준을 적용하였다.

표 4. 알고리즘 적용비율의 선택

적용 알고리즘	증가 비율 γ
algorithm0	
algorithm1	0% ~ 4%
algorithm2	4%~15%
algorithm3	15% ~ 20%
algorithm4	20% ~ 25%
algorithm5	25% ~ 35%
algorithm6	35% ~ 47%
algorithm7	47% ~ 51%
algorithm8	51% ~ 58%

표 3. 제안하는 간이형 복호기의 구현 결과

적용 모듈	실행파일 size		증가되는 조건문 개수 (frame 당)			
	원본 : 228KB	원본대비 증가율	Test1_640x480_2598f	Test2_640x480_3806f	Test3_1280x720_2598f	Test4_1280x720_3806f
MC-luma	260KB	14%	0	0	0	0
MC-chroma	228KB	0%	1265	1345	3666	3335
Deblocking	252KB	10.5%	0	0	0	0
total	288KB	26.3%	1265	1345	3666	3335

V. 모의실험

이 장에서는 실시간 복호화를 만족하는 H.264 동영상 복호기를 구현하여 PC에서 그 성능을 검증하였다. 각 프레임의 복호화가 PTS 이전에 완료가 된 경우에는 실시간성을 만족한 프레임으로 판단하고, 그렇지 않은 경우에는 실시간성을 만족하지 못한 프레임으로 지정해서 실시간성을 만족하는 프레임의 비율(%)로 실시간성의 성능을 측정하였다.

5.1 안정적인 환경

복호화가 이루어지는 동안 같은 수치의 delay를 부여하여 안정적이지만 느린 시스템의 경우를 시뮬레이션 하였다. SD급 영상의 경우 평균 30fps로 복호화 되어지는 실험1에서부터 평균 22fps로 복호화 되어지는 실험5까지 5단계의 경우를 시뮬레이션 하였고(표 5, 표 6), HD급 영상의 경우 평균 25fps ~ 27fps로 복호화 되어지는 실험1에서부터 평균 23fps ~ 24fps로 복호화 되어지는 실험3까지 3단계의 경우를 시뮬레이션 하였다(표 7, 표 8).

표 5. 복호기 검증 결과 (1) (SD급)

실험	성능	기준 디코딩	알고리즘을 사용한 디코딩
1	평균 디코딩 속도 (fps)	30.19	34.83
	실시간을 만족하는 프레임 수	37.45% (973/2598)	99.69% (2590/2598)
2	평균 디코딩 속도 (fps)	28.28	35.21
	실시간을 만족하는 프레임 수	27.33% (710/2598)	99.23% (2578/2598)
3	평균 디코딩 속도 (fps)	26.48	34.80
	실시간을 만족하는 프레임 수	22.67% (589/2598)	98.85% (2568/2598)
4	평균 디코딩 속도 (fps)	23.899	32.89
	실시간을 만족하는 프레임 수	0% (0/2598)	99.00% (2572/2598)
5	평균 디코딩 속도 (fps)	22.71	31.18
	실시간을 만족하는 프레임 수	0% (0/2598)	74.48% (1935/2598)

표 6. 복호기 검증 결과 (2) (SD급)

실험	성능	기준 디코딩	알고리즘을 사용한 디코딩
1	평균 디코딩 속도 (fps)	30.17	36.89
	실시간을 만족하는 프레임 수	43.75% (1665/3806)	99.63% (3792/3806)
2	평균 디코딩 속도 (fps)	28.25	36.03
	실시간을 만족하는 프레임 수	22.99% (875/3806)	99.45% (3785/3806)
3	평균 디코딩 속도 (fps)	26.59	34.59
	실시간을 만족하는 프레임 수	19.92% (758/3806)	99.13% (3773/3806)
4	평균 디코딩 속도 (fps)	23.88	33.11
	실시간을 만족하는 프레임 수	0% (0/3806)	99.19% (3775/3806)
5	평균 디코딩 속도 (fps)	22.67	31.92
	실시간을 만족하는 프레임 수	0% (0/3806)	93.33% (3552/3806)

표 7. 복호기 검증 결과 (3) (HD급)

실험	성능	기준 디코딩	알고리즘을 사용한 디코딩
1	평균 디코딩 속도 (fps)	25.96	37.15
	실시간을 만족하는 프레임 수	27.48% (714/2598)	97.96% (2545/2598)
2	평균 디코딩 속도 (fps)	24.61	35.94
	실시간을 만족하는 프레임 수	25.83% (671/2598)	98.08% (2548/2598)
3	평균 디코딩 속도 (fps)	23.32	34.33
	실시간을 만족하는 프레임 수	23.44% (609/2598)	90.30% (2346/2598)

표 8. 복호기 검증 결과 (4) (HD급)

실험	성능	기준 디코딩	알고리즘을 사용한 디코딩
1	평균 디코딩 속도 (fps)	27.55	37.80
	실시간을 만족하는 프레임 수	27.43% (1044/3806)	97.69% (3718/3806)
2	평균 디코딩 속도 (fps)	26.00	36.79
	실시간을 만족하는 프레임 수	22.02% (838/3806)	98.06% (3732/3806)
3	평균 디코딩 속도 (fps)	24.59	35.61
	실시간을 만족하는 프레임 수	20.49% (780/3806)	97.00% (3692/3806)

SD급 영상의 경우 실험1에서부터 실시간 복호화를 만족하는 프레임이 하나도 없는 경우인 실험4의 상태까지 제안하는 복호기를 사용했을 경우 99% 이상 실시간 복호화를 만족하게 된다. 실시간 복호화가 100%가 되지 않는 이유는 실시간 동작을 판단하고 알고리즘을 변경하여 적용할 때까지의 시간이 필요하기 때문이다.

5.2 불안정한 환경

다음의 경우는 프로세서에서 다른 응용프로그램이 실행되는 등의 돌발적인 상황에 대한 경우로 복호화 중 2번의 환경변화가 있다고 가정하여 시뮬레이션 하였다. 표 9에 기준 복호화 방법을 사용하였을 경우와 제안한 복호기를 사용하였을 경우에 대해 실시간 복호화를 만족하는 프레임 수를 비교하여 나타내었다.

불안정한 환경에서 실시간 복호화를 만족하는 프레임 수가 기준에 20~40%정도 존재할 때, 본 논문에서 제시한 복호기를 이용하였을 경우 96~98%의 실시간 복호화율을 보이는 것을 알 수 있다. 그림 6-9의 결과그래프를 보면 각 테스트 영상마다 1000

표 9. 실시간을 만족하는 프레임 수 (불안정한 환경)

테스트 영상	기준 디코딩	알고리즘을 사용한 디코딩
Test1_640x480_2598f	20.52% (533/2598)	98.46% (2558/2598)
Test2_640x480_3806f	40.96% (1559/3806)	96.93% (3689/3806)
Test3_1280x720_2598f	25.02% (650/2598)	96.84% (2516/2598)
Test4_1280x720_3806f	24.86% (946/3806)	98.00% (3730/3806)

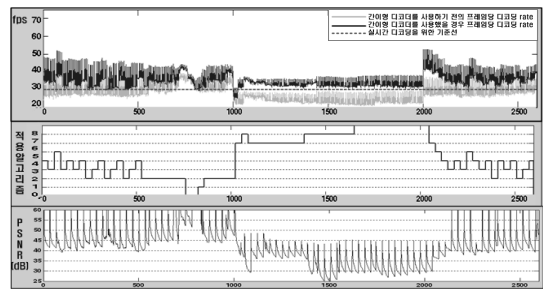


그림 6. 복호기 검증 결과 - 불안정한 환경, Test1(SD급)

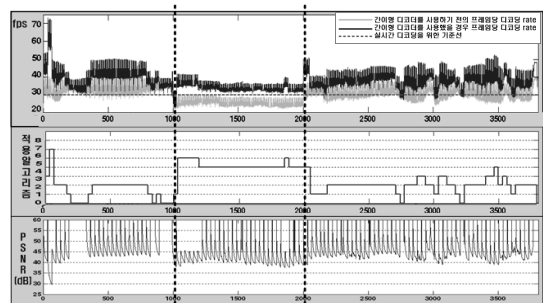


그림 7. 복호기 검증 결과 - 불안정한 환경, Test2(SD급)

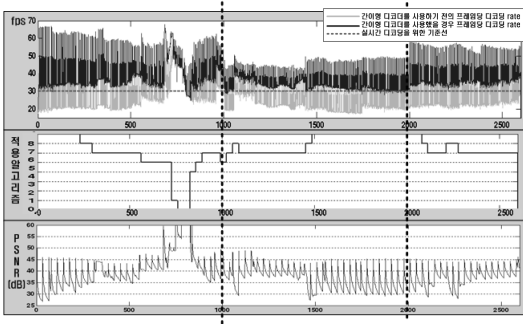


그림 8. 복호기 검증 결과 - 불안정한 환경, Test3(HD급)

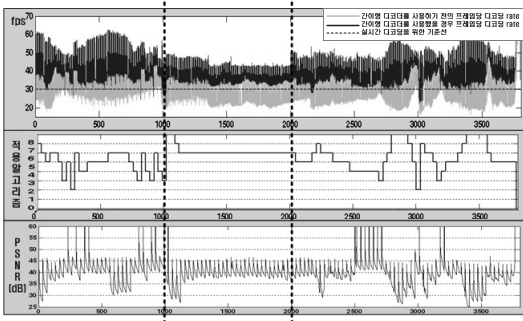


그림 9. 복호기 검증 결과 - 불안정한 환경, Test4(HD급)

번째 프레임에서 환경의 변화가 생기게 되고 그에 따라 적용 알고리즘의 단계가 높아지며, PSNR은 낮아지는 것을 확인할 수 있다.

5.3 실시간 동작이 보장되는데 걸리는 시간

그림 10에는 제안하는 복호기를 적용하였을 경우 실시간 복호화로 전환되기 까지의 시간에 대해 나타내고 있다. 알고리즘의 변경은 1번째 프레임부터 시작하여 33프레임 간격으로 최장 복호화 시간을 체크하여 이루어지게 된다. 1000번째 프레임에서 환경이 변화하였을 경우 1024번째 프레임(1번째 프레임+33

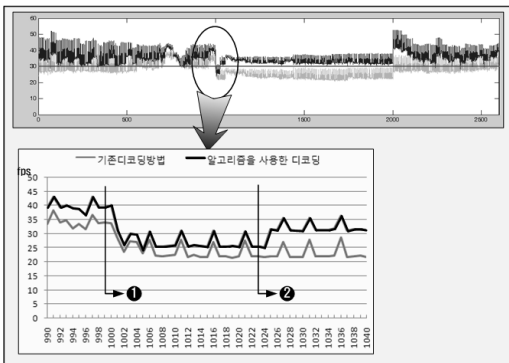


그림 10. 실시간이 보장되는데 걸리는 시간

프레임x31)에서 알고리즘 변경을 판단하게 되고, 바로 다음 프레임부터 변경된 알고리즘이 적용되게 되어 실시간 복호화를 보장하게 된다. 앞에서 99%~95%의 실시간 복호화를 만족하는 이유는 이렇게 일정시간 간격으로 알고리즘 변경을 판단함으로써 실시간 복호화로 전환하는 데 걸리는 시간 때문이다.

VI. 결 론

본 논문에서는 최적화된 FF-MPEG H.264 소스를 기초로 복호화 과정의 일부를 제한하여 실시간복호화를 보장하는 복호기의 구현에 대한 방법을 제시하였다. 먼저 H.264 복호기의 각 모듈에 적용 가능한 15가지 간이형 복호화 방법에 대해 제안을 하고, 이 방법들을 조합한 419개의 알고리즘 중 화질 열화를 최소화 하면서 속도를 향상 시킬 수 있는 8개의 알고리즘을 선택하였다. 이렇게 만들어진 각기 다른 수준의 최적 간이형 복호화 알고리즘은 PSNR을 약 31dB에서 49dB로 유지하면서 원본 복호기의 초당 프레임 수에 비해 최대 59%에서 7%까지 속도를 향상 시킬 수 있다. 이렇게 만들어진 알고리즘들을 하나의 복호기로 만들어 복호화에 적응적으로 실시간 적용시킬 수 있는 효과적인 방법에 대하여 제시하고 모의실험을 하였다. 안정적인 시스템의 경우와 응용프로그램 실행과 같은 돌발적인 상황의 두 가지 경우에 대해 SD급 영상과 HD급 영상을 이용하여 본 복호기에 대한 검증을 하였다. 그 결과 느리지만 안정적인 시스템의 경우, 평균 23~24fps로 돌아가는 환경에서 이 복호기를 사용하였을 경우에 실시간 복호화를 98~99% 정도 만족하였다. 또한 불안정적인 시스템의 경우, 평균 20~40fps로 돌아가는 환경에서 이 복호기를 사용하였을 경우 96~98%정도 실시간 복호화를 만족하는 것을 볼 수 있었다. HD급 영상의 경우에는 범용 컴퓨터에서도 실시간 구현이 어려울 정도로 하드웨어 자원을 많이 필요로 하게 된다. 임베디드 환경일 경우 SD영상과 HD영상의 실시간 재생은 더욱더 힘들어지게 된다. 이러한 경우 본 논문에서 제안한 알고리즘을 적용하면 화질 열화를 최소화하면서 실시간 복호화를 만족하는 환경을 만들 수 있고 제품의 경쟁력을 높일 수 있을 것이다.

참 고 문 헌

[1] “Draft ITU-T Recommendation and Final Draft

international Standard of Joint Video Specification (ITU-T Rec. H.264|ISO/IEC 14496-10 AVC),” *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG*, JVT-G050, Mar. 2003.

[2] FF-MPEG H.264/AVC Software available from <http://svn.mplayerhq.hu/ffmpeg/trunk/>

[3] Ostermann, J.; Bormans, J.; List, P.; Marpe, D.; Narroschke, M.; Pereira, F.; Stockhammer, T.; Wedi, T., “Video coding with H.264/AVC: tools, performance, and complexity,” *IEEE Circuits and Systems Magazine*, Vol.4, Issue1, First Quarter 2004.

[4] Horowitz, M.; Joch, A.; Kossentini, F.; Hallapuro, A., “H.264/AVC baseline profile decoder complexity analysis,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.13, Issue7, July 2003.

[5] X. Quan, L. Jilin, W. Shijie and Z. Jiandong, “H.264/AVC baseline profile decoder optimization on independent platform,” *In Proc. IEEE Wireless Communications, Networking and Mobile Computing*, Vol.2, pp.1253-1256, Sep. 2005.

[6] Iain E.G. Richardson, “H.264 and MPEG-4 Video Compression,” *Wiley*, pp.159-207, 2003.

[7] 방준성, “H.264/AVC 기술동향”, *전자부품연구원*, 2007.

[8] ETRI, “HD급 H.264 기술의 발전 동향”, *ETRI*. 2006.

[9] 정제창 역, “H.264/AVC 동영상 압축 표준”, *홍릉과학출판사*.

김 증 찬 (Jong-chan Kim)

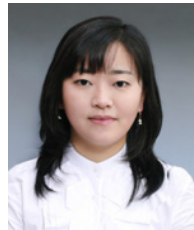
정회원



2005년 2월 한양대학교 전자컴퓨터공학부 졸업
 2007년 2월 한양대학교 전자전기 제어계측공학과 석사
 2007년 2월~현재 LG전자 DTV연구소 주임연구원
 <관심분야> 디지털 영상처리, DMB 시스템, H.264, 영상압축 및 실시간 구현

김 두 리 (Du-ri Kim)

정회원



2006년 2월 한양대학교 전자컴퓨터공학부 졸업
 2008년 2월 한양대학교 전자전기 제어계측공학과 석사
 2008년 2월~현재 삼성전자 DM총괄 VD사업부 연구원
 <관심분야> 디지털 영상처리, H.264, 영상압축 및 실시간 구현

이 동 호 (Dong-ho Lee)

정회원



1986년 2월 한양대학교 전자공학과 졸업
 1988년 12월 Texas 대학 전기공학과 석사
 1991년 5월 Texas 대학 전기공학과 박사
 1991년 6월~1994년 2월 LG전자 영상미디어연구소 선임연구원
 1994년 3월~현재 한양대학교 전자 및 통신 공학과 교수
 <관심분야> 디지털 영상처리, 디지털 TV, 영상 압축 및 실시간 구현