

모바일 기기를 위한 H.264 인코더의 최적 매개변수의 결정

류민희¹, 김형신^{1*}
¹충남대학교 컴퓨터공학과

Optimal Parameter Selection of H.264 Encoder For Mobile Devices

Minhee Ryu¹ and Hyungshin Kim^{1*}

¹Department of Computer Science and Engineering, Chungnam National University

요 약 최근 스마트폰과 태블릿과 같은 소형 모바일 기기가 확산됨에 따라, 모바일 기기에서 동영상 촬영시 이용되는 비디오 인코더의 성능 최적화의 필요성이 제기되고 있다. 본 논문에서는 모바일 기기를 대상으로 하는 H.264/AVC 기본 프로파일 비디오 인코더를 모바일 기기에 구현하고, 실험을 통해 H.264 인코더의 주요 제어 변수를 최적화하였다. 실험으로는 인코더의 복잡도에 영향을 주는 것으로 알려진 라그랑지안 최적화, 하다마드 변환, 움직임 벡터 탐색 범위, I-프레임 주기, 참조 프레임 수를 다양하게 조합하여 변화시키면서 동영상의 화질, 비트율, 인코딩 시간, 움직임 추정 시간 그리고, 인코딩에 따라 보드에서 소모되는 전력을 측정하였다. 실험에서 측정된 데이터를 분석하여, 모바일 기기에서의 비디오 인코더에서 요구되는 조건을 만족할 수 있는 최적의 H.264/AVC 제어 변수를 위의 다섯 가지 기능 모듈에 대해 결정하였다.

Abstract As many mobile devices such as smart phones and tablets are widely spread, optimized mobile video encoder used during video recording application is needed. In this paper, we implemented H.264/AVC base profile video encoder on a mobile device and empirically optimized control parameters of the encoder. As the experiment, we more than 100 test cases were designed with varying Lagrangian optimization, Hadamard Transform, search range, I-frame period, and reference frames. During the experiment, we measured picture quality, bit-rate, encoding time, motion estimation time, and power consumption. From the result, we can determine optimal values for the H.264 control parameters.

Key Words : Video encoder, Mobile video, H.264, Encoder optimization, Energy consumption

1. 서론

최근 스마트폰, 태블릿 PC등과 같은 모바일 기기가 널리 보급되고 무선 컴퓨팅 환경이 일반화되면서, 멀티미디어 콘텐츠에 대한 시장의 요구가 점차 증가하고 있다. 기존의 동영상 압축 표준안인 MPEG-4와 H.263등은 화질과 압축률에서 한계를 드러내기 시작했고, 그 대안으로 H.264/AVC가 표준화되었다[1]. H.264/AVC는 이전 표준안들과 비교했을 때 동일한 해상도와 화질에 대해 약 50% 개선된 압축 성능을 보이고, 동일 전송속도를 기준

으로 MPEG-4보다 2배 정도 높은 화질을 보장한다[2]. 또한, 네트워크 전송률의 변화에 대하여 높은 적응성을 가지며, 이러한 특성은 화상통화나 화상회의 등 실시간 인코딩을 필요로 하는 서비스에 특히 필요하다.

H.264/AVC는 기존 표준들과 비교했을 때 전송률 왜곡(Rate-Distortion)에 있어 주목할 만한 향상을 가져왔다. 하지만, 기존 표준들에 비해 높은 계산 복잡도로 전용 하드웨어 추가 없이 모바일 기기의 제한된 리소스만으로 실시간(real-time) 또는 사용자가 만족할 수 있는 시간 내에 처리가 가능한 H.264 코덱(codec)을 구현하기는 어렵

이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업입니다.(No. 2012-0004576)

*Corresponding Author : Hyungshin Kim

Tel: +82-10-2085-3860 email: hyungshin@cnu.ac.kr

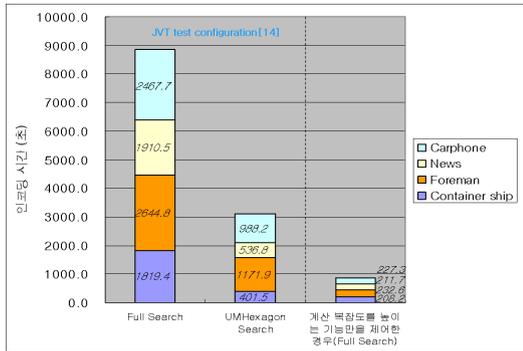
접수일 12년 07월 20일

수정일 (1차 12년 08월 22일, 2차 12년 09월 07일)

게재확정일 12년 10월 11일

대[3]. 하지만, 최근모바일 기기의 하드웨어 성능 향상으로 모바일 기기에서의 H.264/AVC 적용에 대한 연구가 차츰 진행되고 있다[4,5].

기존 연구에서는 주로 실행 시간을 줄이기 위해 새로운 알고리즘들을 제시하였다. 하지만, H.264/AVC 인코더의 계산 복잡도를 높이는 기능 자체에 대한 분석을 수행하고, 이들의 실행 매개변수를 조절한다면 일정 수준의 성능 개선을 얻을 수 있다.



[그림 1] 매개 변수 조절을 통한 인코딩 속도 개선
[Fig. 1] Encoding time after parameter selection

그림 1은 기본 움직임 검색 알고리즘인 Full Search와 개선된 UMHexagon Search 방식의 움직임 추정 알고리즘을 적용한 후, 실제 모바일 기기에 구현 인코딩 시간을 측정한 것이다. 가장 우측의 그래프는 본 연구의 결과로 결정된 최적 매개변수 값을 적용한 후에 Full Search 방식을 사용하는 인코더의 성능을 측정한 것이다. UMHexagon Search 방식의 경우가 Full Search 보다 빠른 인코딩 시간을 보여주고 있지만, 새로운 성능개선 알고리즘의 도입이 없이, 계산 복잡도를 높이는 제어 매개변수를 Full Search에 대해 최적화한 것이 가장 빠른 인코딩 성능을 보여주는 것을 알 수 있다.

따라서, 이 논문에서는 성능 향상을 위해 최적화 알고리즘을 적용하기에 앞서, 인코더의 계산 복잡도를 높이는 기능들에 대한 제어를 통해 모바일 기기에 적합한 화질 및 인코딩된 비트율의 저하가 거의 없는 빠른 인코딩을 수행할 수 있음을 보이고자 한다.

본 논문에서는 모바일 기기를 대상으로 하는 H.264/AVC 기본 프로파일 비디오 인코더를 실제 모바일 기기에 구현하고, 인코더의 부하(overhead)를 높이는 기능들을 찾아 제어함으로써 화질, 비트율, 인코딩 시간, 움직임추정 시간 그리고, 인코딩에 따라 보드에서 소모되는 전력을 측정하였다. 측정된 데이터를 분석하고,

H.264/AVC에 대한 최적화 알고리즘 적용 전 화질 및 인코딩 비트율의 큰 저하 없이 인코딩 시간 및 전력 소모량이 최소인 기준에 맞는 제어 변수를 결정하였다.

본 논문의 구성은 다음과 같다. 2장에서는 모바일 기기에서의 동영상 인코더 성능 최적화와 관련된 기존의 연구를 분석한다. 3장에서는 H.264/AVC 인코더의 성능 측정 방법을 설명하였다. 4장에서는 3장의 실험 설정을 이용하여 주요 기능에 대한 성능을 측정하고, 최적의 성능을 보이는 매개변수 값을 결정한다. 5장에서는 결론을 맺는다.

2. 관련연구

이 절에서는 모바일 기기의 특징에 맞는 비디오 인코더의 성능 개선 연구들을 설명한다. 모바일 시스템의 소프트웨어는 작은 공간에 다수의 하드웨어가 집적되어야 하므로, 프로세서의 능력 또는 메모리, I/O 장치 등의 사용에 있어 제약이 따른다. 모바일 기기는 자원 제약으로 인해 데스크 탑 급 컴퓨터 이상의 성능을 발휘하기 힘들며, 최대 성능을 유지하기 위해서는 그만큼의 전력 소모가 뒤따른다.

최근 들어 스마트 폰과 태블릿과 같은 모바일 기기들이 필수품처럼 되었으며, 멀티미디어에 대한 관심과 함께 성능 향상 요구가 증가하고 있다. 그와 더불어 기존 코덱에 비해 성능이 우수한 H.264/AVC를 모바일 기기에 적용한 사례들이 발표되고 있다. 하지만, 주로 디코더 위주의 연구 또는 하드웨어로 구현된 연구가 대부분이다[6,7]. [4]와 [8]은 소프트웨어로 구현된 H.264 비디오 인코더의 성능을 개선한 연구들이다. 이 들 논문에서는 H.264 비디오 인코더의 계산 복잡도를 높이는 인트라 프레임 예측 모듈, 움직임 추정 모듈, 변환 및 양자화 모듈 전반에 걸친 최적화를 통해 H.264 기본 프로파일(Baseline Profile)을 모바일 기기에 적용하였으며, 움직임 추정 모듈과 모드 선택 모듈에 대한 최적화를 통해 H.264 비디오 인코더 전반의 성능을 향상 시켰다[4,8].

인코더와 전력소모량과 관련된 연구로는 DMB(Digital Multimedia Broadcast), DVB-H(Digital Video Broadcast for Handheld)와 같은 휴대용 단말의 저전력 하드웨어 구현이 대부분이다[7]. 이들 기존 연구에서는 최적화 이후 검증을 위한 H.264 인코딩 기능들에 대한 제어 변수를 선정 방법이 명확하지 않다.

따라서 본 논문에서는 모바일 기기에 적합한 최적의 인코딩 기능 제어 변수를 찾기 위해 5가지 매개 변수 값의 설정을 변경하여 144개의 실험 항목을 만들고, 성능을

측정하여, 최적의 성능을 보이는 제어 값을 결정하였다.

3. H.264 인코더 성능분석 방법

이 장에서는 H.264의 계산 복잡도를 증가시키는 기능들 선정하고, 각 기능과 인코딩 성능 지표들간의 관계를 분석할 수 있는 실험 방법에 대해 설명한다.

성능 분석 실험을 위해 타겟 보드에 리눅스(Linux) 2.4.20 커널(Kernel) 버전을 설치하였다. H.264 코덱으로는 JM11.0을 사용하였다.

성능 측정을 위한 모바일 타겟은 모바일 기기 개발용 플랫폼인 SMDK2440A Evaluation Board[10]를 사용했다. 프로세서는 ARM920T 코어 기반의 삼성 S3C2440A 마이크로 프로세서를 사용하고 있다. 프로세서는 400MHz의 클럭속도로 동작하도록 설정되어 있으며, 16KB 64-way 인스트럭션/데이터 캐시(associative instruction/data cache)를 사용한다.

인코더 실행 시 전력 측정을 위해 FLUKE 사의 멀티미터를 사용했다[11]. 보드에 연결되는 전원은 멀티미터(multimeter)를 거쳐 보드에 전달되고, 멀티미터기로 입력되는 전류값을 측정할 수 있다. 초당 샘플링이 가능하고, 시리얼(serial)로 컴퓨터에 연결하여 측정된 데이터를 컴퓨터에 로그(log) 파일로 생성해준다.

H.264 비디오 인코더는 프레임 내 예측(Intra-frame Prediction), 프레임 간 예측(Inter-frame Prediction), 변환 및 양자화(Transform & Quantization), 디블록킹 필터(Deblocking Filter), 엔트로피 코딩(Entropy Coding) 등의 주요 모듈로 구성되어 있다. 기본 프로파일(Baseline Profile)에서 다음 5가지 기능들이 계산 복잡도를 높이는 요인으로 알려져 있다[3].

(1) 라그랑지안(Lagrangian) 전송률 왜곡 최적화

SAD(Sum of Absolute Differences), PSNR(Peak Signal to Noise Ratio) 등 영상의 왜곡 정도를 측정하기 위해 라그랑지안 계수를 사용하는 곱셈기를 사용함으로써 전송률 왜곡 성능을 향상시킬 수 있다. 이를 통하여 화질 및 비트율이 각각 0.35dB와 9% 향상시킬 수 있으나, 인코더 실행 시간이 최대 120% 길어질 수 있다.

(2) 하다마드 변환(Hadamard Transform)

'1'과 '-1'만으로 구성된 직교 행렬(orthogonal matrix)를 이용하여 휘도 DC(Luma DC) 신호를 위한 4x4 화소단위 이산 하다마드 변환(4x4 Discrete Hadamard Transform)과

색차 DC(Chroma DC) 신호를 위한 2x2 화소단위 이산 하다마드 변환(2x2 Discrete Hadamard Transform)을 수행한다. 이는 인코딩된 비트율에 대해 성능 향상을 보인다.

(3) 움직임 벡터 탐색 범위(Motion Vector search range)

움직임 벡터를 찾기 위한 탐색 범위를 의미하며, 범위가 넓을수록 보다 정확한 움직임 벡터를 찾을 수 있으므로 압축률을 높일 수 있으나, 넓은 범위에 걸쳐 탐색을 해야 하므로 계산량이 많아진다. 1개의 참조 프레임을 사용할 때 약 8 픽셀 범위와 32 픽셀 범위에 대해 약 9배의 계산 복잡도 차이가 발생한다[3]. 본 논문에서 수행한 성능 측정 실험에서도 이 항목이 성능에 가장 큰 영향을 미치는 것을 확인할 수 있었다.

(4) I-프레임 주기

H.264 기본 프로파일에서는 I-프레임과 P-프레임을 사용한다. I-프레임은 원본 영상에 가장 가깝게 압축하며, P-프레임은 압축이 완료된 참조 프레임과 원본 영상의 차이를 이용하여 압축한다. P-프레임이 많을수록 비트율은 낮아지지만 영상의 화질이 떨어진다. I-프레임이 많으면 많을수록 원본 영상에 가까운 화질을 갖지만, 압축의 효율은 떨어지게 된다. 본 논문에서는 레벨1, 1b 기준에 적합한 I-프레임 주기를 결정한다.

(5) 참조 프레임 수

참조 프레임이 많을수록 가장 정확한 움직임 벡터를 찾을 수 있다. 이는 압축률을 높이는 효과가 있다. 하지만, 참조 프레임의 수가 많아질수록 비교를 위한 메모리 공간이 추가적으로 필요하며, 비교 연산 횟수가 많아지므로 전체 실행 시간 성능에 영향을 미친다.

[표 1] 계산 복잡도가 높은 기능별 제어 조건
[Table 1] Control parameters for complex operation in H.264

기능	제어 조건
라그랑지안 전송률 왜곡 최적화	On / Off
하다마드 변환	On / Off
움직임 벡터 탐색 범위	8픽셀 / 16픽셀 / 32픽셀
I-프레임 주기	오직 한번만 / 6 프레임 / 12 프레임
참조 프레임 수	4 / 3 / 2 / 1

최적의 인코딩 성능을 위한 제어조건을 결정하기 위하여 위와 같은 H.264의 계산 복잡도를 증가시키는 5가지 기능들에 대해 표 1의 제어 조건을 조합한 총 144개 성능 측정 테스트 세트를 생성하였으며, 각 항목은 ‘Case_000’로 명명하였다.

테스트를 위한 동영상으로 대표적인 시험용 동영상 중에서 움직임 변화가 적은 Container Ship과 움직임 변화가 많은 Foreman을 사용했다. 각각의 테스트 세트를 적용한 H.264 인코더를 두 개의 동영상에 대해 타겟 보드에서 실행시키면서 화질, 인코딩된 비트율, 인코딩 시간, 움직임 추정 시간 그리고, 보드의 전력 소모량을 측정했다. 전력 소모를 제외한 나머지 결과들은 인코더 실행 시 자동 생성되지만 보드의 전력 소모량을 측정하기 위해서는 추가적인 작업이 필요하다. 에너지 소모량은 멀티미터기를 이용하여 측정된 값으로부터 식(1)과 같이 계산하였다.

$$E = V_{measured} \times \sum^T (I_{measured} - I_{norm}) \quad (1)$$

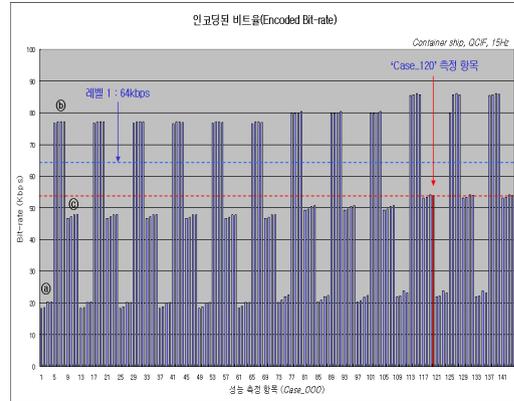
멀티미터기를 이용하여 매초 간격으로 보드에 입력되는 전류값을 측정했다. 인코딩을 위해 소모된 에너지는 초당 측정된 전류 값($I_{measured}$)에서 평상시 보드에서 사용되는 평균 전류(I_{norm}) 값을 뺀 나머지를 인코딩 시간 T 동안 누적하고, 입력 전압($V_{measured}$)을 곱하여 구할 수 있다.

4. H.264 인코더 최적인자 결정

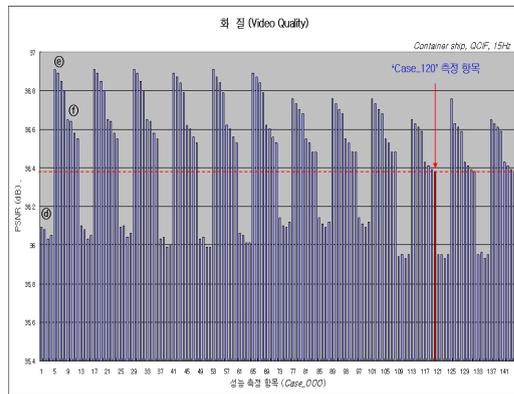
이번 절에서는 3절의 실험 방법을 사용하여 얻은 결과를 설명한다. [그림 2]는 비트율 성능 측정 결과이다. X축은 측정 항목 144개의 테스트 세트 번호이며, Y축은 비트율을 표시하고 있다. 비트율 값의 변화는 I-프레임 주기에 따라 ㉠는 100 프레임 마다 I-프레임이 1개만 존재하는 경우, ㉡는 6 프레임 마다 1개의 I-프레임, ㉢는 12 프레임 마다 1개의 I-프레임 주기를 갖는 경우이다. Case_73 ~ 144 항목은 라그랑지안 전송을 왜곡 최적화를 사용하지 않는 경우이다. 이 경우 하다마드 변환을 사용하므로 인코딩된 비트율이 낮아지는 효과를 확인할 수 있다.

그림 3은 인코딩 후 화질에 대한 성능 측정 결과이다. X축은 측정 항목 144개의 케이스 번호이며, Y축은 PSNR 값을 표시하고 있다. 전체적으로 비슷한 결과값을 보이나, I-프레임 주기의 차이에 따라 약간씩 차이를 보인다. ㉠는 전체 인코딩에 있어 오직 처음 한번만 I-프레

임 인코딩을 하고, ㉡는 6 프레임 마다 한 프레임을 I-프레임으로 인코딩 하고, ㉢는 12 프레임 마다 한 프레임을 I-프레임으로 인코딩한다. ㉠ > ㉡ > ㉢ 순으로 화질이 좋은 모습을 보여주고 있다.



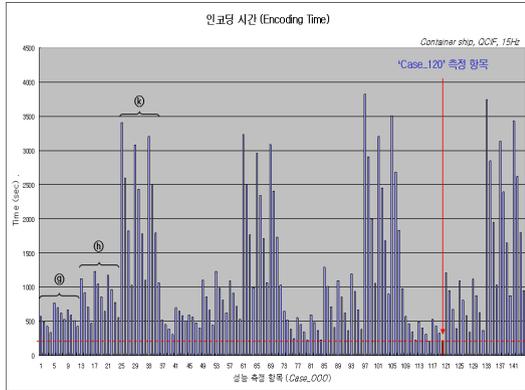
[그림 2] 비트율 성능 측정
[Fig. 2] Bit-rate measurement



[그림 3] 화질 성능 측정
[Fig. 3] PSNR measurement

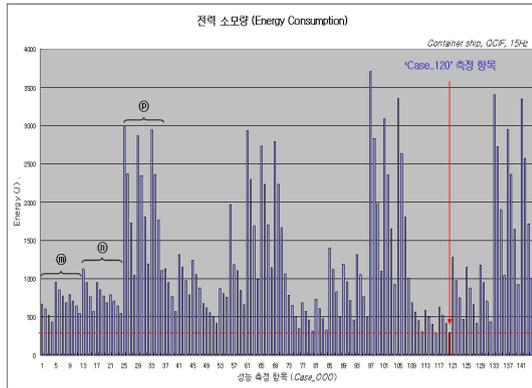
그림 4는 총 인코딩 시간에 대한 성능 측정 결과이다. X축은 측정 항목 144개의 케이스 번호이며, Y축은 총 인코딩 시간을 표시하고 있다. 인코딩 시간은 탐색 범위 (search range)의 영향이 크다. ㉠는 8 픽셀, ㉡는 16 픽셀, ㉢는 32 픽셀 탐색 범위에 따른 총 인코딩 시간 차이를 보여준다. ㉠, ㉡, ㉢ 테스트 케이스들의 내부적으로 인코딩 시간 차이를 보이는 것은 I-프레임 주기와 참조 프레임 수의 영향이다. 이들을 각각 살펴보면, 내부적으로 4개의 테스트 세트마다 반복되는 성능 차이는 참조 프레임의 영향으로 나타나는 것이며, I-프레임 주기에 따라서

도 인코딩 시간 차이를 보인다. 이는 I-프레임을 인코딩하는 시간이 P-프레임 인코딩 시 계산 복잡도가 높은 움직임 추정 모듈을 실행하는 시간보다 인코딩 전체로 볼 때 더 많은 시간이 소모된다는 것을 의미한다.



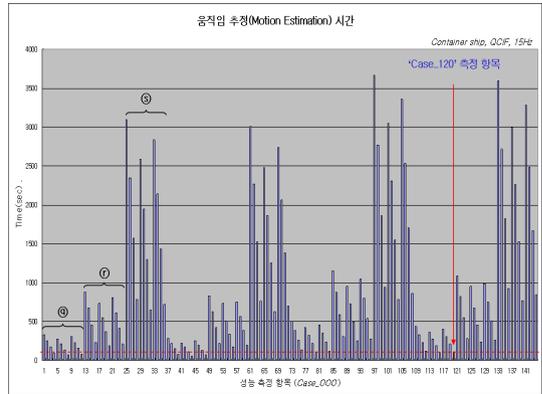
[그림 4] 인코딩 시간 측정 결과
[Fig. 4] Encoding time measurement

그림 5는 에너지 소모량 측정 결과이다. Y축은 전력 소모량을 표시하고 있다. 소모된 에너지는 탐색 범위의 영향이 크다. ㉓은 8 픽셀, ㉒은 16 픽셀, ㉑는 32 픽셀 탐색 범위에 따른 전력 소모량 차이를 보여준다.



[그림 5] 에너지 소모량 측정
[Fig. 5] energy consumption measurement

그림 6은 움직임 추정에 소요된 총 시간을 표시하고 있다. 그림 4, 그림 5와 마찬가지로 탐색 범위의 영향이 가장 많이 받는다. ㉑는 8 픽셀, ㉒은 16 픽셀, ㉓는 32 픽셀 탐색 범위에 따른 총 소요 시간 차이를 보여준다.



[그림 6] 움직임 추정 시간 측정결과
[Fig. 6] Motion estimation time measurement

위와 같이 실행 성능 측정 결과 그림 2, 그림 3은 I-프레임의 영향을 많이 받으며, 그림 4, 그림 5, 그림 6은 움직임 벡터 탐색 범위의 영향을 받고 있다. 이는 움직임 벡터를 찾기 위해 얼마나 많은 계산 횟수를 반복하느냐의 차이로 인한 결과이다. 그리고 X축 4개의 테스트 케이스마다 유사한 패턴의 결과 값이 발생하는 것은 참조 프레임 수에 따른 것이다. 화질의 경우 최대/최소 PSNR 값의 차이는 0.98(dB)로 큰 차이를 보이지 않았다.

모바일 기기의 요구사항을 만족할 수 있는 인코딩 제어 값을 선정하기 위해서는 다음의 기준을 만족해야 한다. 첫째, 실행 시간 및 전력 소모가 최소화 되어야 한다. 둘째, 비트율이 레벨 1.0을 만족해야 한다. 셋째, 예러에 강하기 위하여 I-프레임의 수가 많아야 한다. 제시한 세 가지 기준에 따라 상위 3개의 후보를 선정하였다. 실행 시간 및 전력 소모에 대해 Case_116, Case_120, Case_80이 선정되었다. 이 중에서 Case_116, Case_80 두 항목은 비트율이 레벨 1.0을 초과하여 제외되고, 나머지 Case_120 항목이 성능이 가장 우수한 것으로 선정됐다.

이 결과들을 종합하면, 최소 실행 시간, 저전력 그리고, 레벨 1.0 기준 비트율 64kbps를 모두 만족하는 테스트 케이스는 Case_120이다. 표 2는 Case_120 측정 항목의 설정값과 성능 측정 결과를 보여준다. 선택된 테스트 케이스 120번은 인코딩 시간이 오래 걸리는 연산들을 최소화하면서 인코딩 품질과 성능을 유지할 수 있는 변수 설정을 갖는 것으로 볼 수 있다. 따라서, 모바일 기기에서 H.264 인코더를 실행하기 위해서는 [표 2]와 같은 제어 값을 설정하면 최적의 성능을 얻을 수 있다.

[표 2] Case_120의 설정값 및 성능측정 결과
 [Table 2] Control values of Case_120 and its performance

측정 항목	Case_120	RD 최적화	Off
하다마드 변환	Off	움직임 벡치 탐색범위	8픽셀
I-프레임 주기	12	참조 프레임	1
비트율(kbps)	53.9		
화질(dB)	36.4		
인코딩 소요시간 (sec)	208.0		
전력소모(Joule)	295.4		
움직임 추정 소요시간(sec)	109.9		

5. 결론

본 논문에서는 모바일 기기에 H.264 기본 프로파일 비디오 인코더를 적용하기 위해 실행 시간 및 전력 소모를 줄이는 방법에 대해 살펴보았다. 이를 위해 H.264 비디오 인코더의 실행 성능에 영향을 미치는 계산 복잡도가 높은 기능들의 제어 변수들을 선정하고, 다양한 제어 값을 적용하여 실제 하드웨어에 구현함으로써, 최적의 성능을 내는 제어 값을 발견하였다. 본 논문에서 실험을 통하여 결정한 최적 설정 조건을 이용하면, 향후 향후 모바일 멀티미디어 연구자들 또는 개발자들이 모바일 기기에 H.264를 적용하기 위한 인코더 최적화 연구에 대해 도움을 줄 것으로 기대한다.

References

[1] T. Wiegand, G.J. Sullivan, G. Bjntegaard, A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, Issue 7, Jul. 2003, pp. 560-576

[2] N. Kamaci, Y. Altunbasak, "Performance comparison of the emerging H.264 video coding standard with the existing standards," Proceedings. 2003 International Conference on Multimedia and Expo, Vol. 1, July 2003, pp. I-345-8 vol. 1

[3] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, T. Wedi, "Video coding with H.264/AVC: tools, performance, and complexity," IEEE Circuits and Systems Magazine, Vol. 4, Issue 1, First Quarter 2004, pp. 7-28

[4] G. Nageswara Rao, Prasad RSV, D. Jaya Chandra and Sridvidya Narayanan, "Real-Time Software Implementation of H.264 Baseline Profile Video Encoder for Mobile and Handheld Devices," 2006 IEEE International

Conference on Acoustics, Speech and Signal Processing, Vol. 5, May 2006, pp. V-457-V-460

[5] Iain E. G. Richardson, "H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia," John Wiley&Sons Ltd, ISBN: 0-470-84837-5, 2003

[6] Tsu-Ming Liu, Ting-An Lin, Sheng-Zen Wang, Chen-Yi Lee, "A low-power dual-mode video decoder for mobile applications," IEEE Communications Magazine, Vol. 44, Issue 8, Aug. 2006, pp. 119-126

[7] B. Stabernack, H. Hubert, K.-I. Wels, "A H.264 video coprocessor for mobile DVB-H terminals," International Conference on Consumer Electronics 2006, Digest of Tech. Papers, Jan. 2006 pp. 89-90

[8] K. Udyawar, A. Ramachandran, S. Shandilya, "Motion estimation and mode decision for low power H.264," International Conference on Consumer Electronics 2006, Digest of Tech. Papers, Jan. 2006, pp. 31-32

[9] JVT(Joint Video Meeting) <http://ftp3.itu.ch/av-arch/jvt-site/>

[10] Meritech co., <http://www.meritech.co.kr>

류 민 희(Minhee Ryu)

[정회원]



- 2007년 2월 : 충남대학교 컴퓨터 공학과 (석사)

<관심분야>

내장형 시스템, 시스템 소프트웨어, 멀티미디어

김 형 신(Hyungshin Kim)

[정회원]



- 1990년 12월 : Univ. of Surrey, 위성통신공학 (석사)
- 2003년 2월 : 한국과학기술원 전산학과 (박사)
- 2003년 4월 ~ 2004년 2월 : Carnegie Mellon Univ. 박사후 연구원
- 2004년 2월 ~ 현재 : 충남대학교 부교수

<관심분야>

내장형 시스템, 시스템 소프트웨어, 우주용 컴퓨터