

DMB 서비스의 비디오 규격, H.264

□ 이철수* / *LG전자 DTV연구소

I. 서론

2005년 상반기 실시 예정인 DMB(Digital Multimedia Broadcasting)서비스는 이동환경 하에서 수신할 수 있는 방송의 한 형태이다. Eureka-147[1]이 정의하는 스트림 모드에 추가적인 채널코딩 기법을 추가해서 기존의 DAB보다 낮은 에러율을 확보하고, DMB는 그 채널을 통해서 비디오, 오디오, 데이터를 하나의 서비스로 전송할 수 있는 비디오 서비스이다. 이 서비스에서 정의하는 비디오의 규격은 ITU-T H.264/ISO/IEC 14496-10 AVC(이하 H.264)[2]를 채택하고 있으며, 오디오 규격은 ISO/IEC 14496-3 ER-BSAC[3], 데이터 규격은 ISO/IEC 14496-11[4]에서 정의하는 시스템 규격을 사용하고 있다. 여기서, 오디오 규격은 기존의 Eureka-147에서 정의하고 있는 MUSICAM과는 별도로 비디오와 같이 전송하며, 오디오 규격을 받

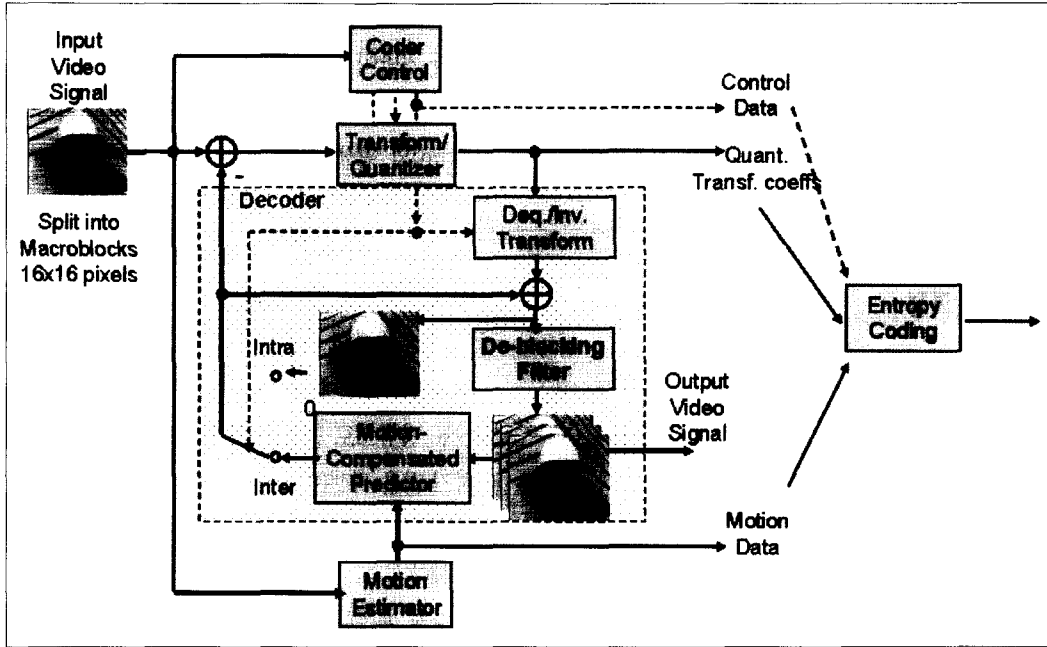
한다.

이 원고에서는 DMB서비스에서 채택하고 있는 비디오 규격인 H.264를 DMB에서 사용하는 부분을 중심으로 살펴보기로 한다.

II. H.264의 개요 [5]

H.264는 기본적으로 블록단위로 영상을 압축하는 기법의 일종이다. 기본적인 블록다이어그램은 <그림 1>에 보인 바와 같다.

전체적으로는 대부분의 압축기술과는 유사하지만, 각각의 기술들을 살펴보면 차이점들이 존재한다. H.264에서 사용한 코딩 기법중에서 기존의 MPEG-2 [6] 혹은 MPEG-4 Part 2 [7] 대비 향상된 기법에 대해서 간략하게 짚고 넘어가기로 한다.



〈그림 1〉 H.264의 베이직 블록 다이어그램

예측기법의 향상을 위해서 다음과 같은 기법들을 사용하였다.



- 움직임 추정에서 사용되는 블록의 크기를 4x4 픽셀 단위 블록까지 사용
- 움직임 추정에서 1/4 픽셀 단위의 해상도
- 영상의 범위를 넘어서는 움직임 벡터를 사용
- 움직임 추정에서 여러 장의 레퍼런스 영상을 사용
- 영상의 전체적인 어두워짐 혹은 밝아짐에 대해서 가중치 예측
- "Skip"과 "Direct" 모드를 이용
- Intra모드의 예측에서 공간적 영역(spatial domain)에서 예측을 수행
- 디코딩 루프안에 위치시킨 deblocking 필터

비디오 코딩의 효율을 개선시키기 위해서는 다음과 같은 기법들을 사용하였다.



- 작은 크기의 블록 변환 (4x4 Integer transform)
- 계층구조의 블록 변환
- Short word 길이의 변환
- 역변환에 있어서의 불일치 해결
- 산술적 코딩 기법의 도입
- 컨텍스트 적응적 엔트로피 코딩 기법의 도입

에러 환경에 대비해서 적용한 기법은 다음과 같다.



- 매개변수에 관한 데이터 구조를 적용
- 문법의 체계를 전송방법에 독립적으로 기술
- 슬라이스의 크기에 대한 자유도
- 매크로블록 순서의 임의성 도입
- 슬라이스 순서에 임의성 도입
- 군더더기 영상의 도입
- SP/SI 영상의 도입

Ⅲ. 비디오 압축 계층의 특징

이 장에서는 크게 예측과정, 양자화와 변환방법, 엔트로피 코딩 방법, 그리고, 디블록킹 과정에 대해서 각 절에서 설명하도록 한다.

1. 예측 과정

H.264에서는 크게 두 가지 기법의 예측방법을 사용한다. 하나는 Intra prediction이며, 나머지 하나는 Inter prediction이다.

1) Intra prediction

Intra 프레임을 구성하는 매크로블록의 타입은 I_PCM, Intra_16x16, Intra_4x4의 세가지가 있다. Inter 프레임에서도 Intra prediction을 사용할 수 있다. luma와 chroma 픽셀에 대해서 유사한 방법으로 Intra prediction을 수행한다.

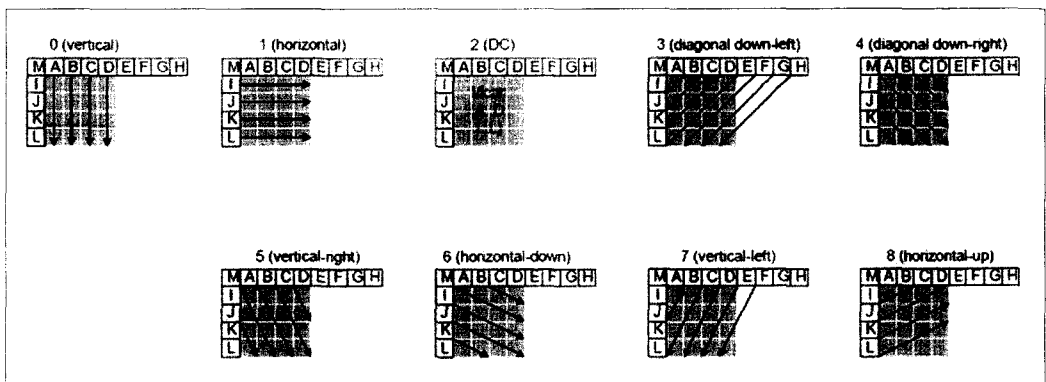
H. 264에서 사용하는 intra prediction은, 기존의 MPEG-4 Visual에서 사용하는 intra prediction과

는 다른 방법을 사용한다. MPEG-4 Visual에서는 예측의 기반이 되는 값으로 변환 영역의 계수 값을 사용하지만, H.264에서는 공간 영역(spatial domain)의 픽셀 값을 사용한다. H.264에서 루프 안으로 들어온 deblocking 필터의 결과값을 사용하지 않고, 그 이전의 값을 이용한다.

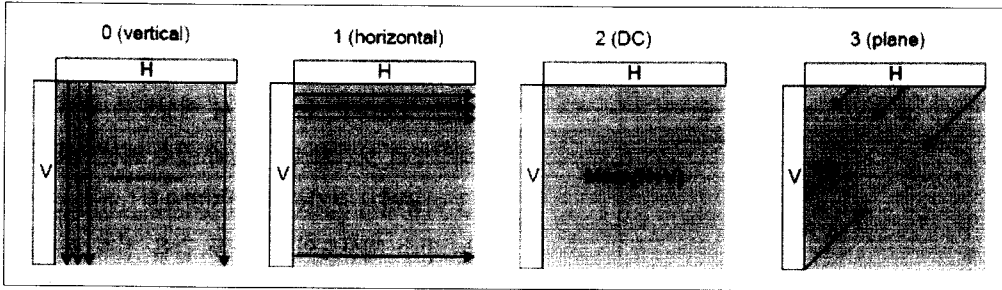
① Intra_4x4의 예측 방법

Intra_4x4의 예측방법을 <그림 2>에 나타냈다. 화살표 방향으로 값들을 예측하며, 만일 아직 디코딩하지 않은 픽셀(예를 들어, E,F,G,H의 경우)을 참조해야 할 경우에는 D의 값을 동일하게 반복시켜서 사용한다.

이렇게 예측한 값과 원 영상의 차분 값을 변환과정을 거쳐서 수신단으로 전송하는데, 이 비트열의 순서는, 예측 방법의 모드가 차분 값의 데이터보다 먼저 전송된다. 따라서, 예측 모드를 모두 디코딩하고, 차분 값들을 다시 디코딩을 하면서 예측과정을 수행해야 한다. 이로 인해 intra_4x4 모드의 경우, 예측과정과 변환과정을 병렬적으로 진행할 수 없다.



<그림 2> Intra_4x4의 예측방법



〈그림 3〉 Intra_16x16의 예측 모드

② Intra_16x16 예측 방법

Intra_16x16 예측 모드는 매크로블록 타입에 포함되어서 전송한다. 네 가지의 경우가 있으며, 각각의 예측 방법은 〈그림 3〉에 나타났다. 각 매크로블록에서 인접하는 주변 픽셀 값 (디블록킹 필터링 이전의 값)을 이용해서 현재 디코딩하는 매크로블록의 값을 예측한다.

③ Chroma 예측 방법

크로마의 경우에는 Intra_16x16 예측 방법과 유사하다. DC 예측이 예측 모드 '0'으로 설정되는 점과, chroma의 경우에는 intra prediction의 단위는 항상 8x8의 형태만을 지닌다는 점이 다른 점이다. Chroma의 예측 모드는 chroma의 두 성분, 즉 Cb와 Cr에 공통적으로 적용해서 예측을 수행한다.

2. Inter prediction

Inter prediction은 이미 디코딩된 참고 영상로의 움직임 벡터를 이용해서 현재 디코딩하는 블록의 값을 예측하는 방법이다. 이를 위해서는 참고 영상의 인덱스, 움직임 벡터, 그리고, 현재 디코딩하는 매크로블록의 형태에 관한 정보들이 필요하다.

1) 매크로 블록의 형태

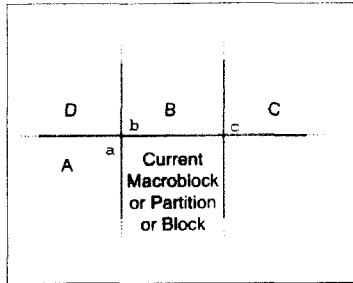
P 매크로블록은 어떻게 파티션하는가에 따라서, 움직임 벡터의 개수가 정해진다. 하나의 매크로블록이 하나의 파티션으로 되는 경우, 둘로 되는 경우, 넷으로 되는 경우가 있으며, 넷으로 되는 경우는 각각이 다시 하나, 둘, 혹은 넷으로 나뉠 수 있다. 따라서, 하나의 매크로블록을 가장 많이 파티션하는 경우는 열 여섯개의 블록으로 나뉘어질 수 있으며, 각각의 블록은 개별적인 움직임 벡터를 가진다.

2) 참고 영상의 인덱스

H.264에서 기존의 비디오 압축 알고리즘과 다르게 참고 영상을 여러 장을 사용한다. 따라서, 움직임 벡터가 참고하는 영상이 어느 영상인지에 대해서도 알려주어야 한다. 참고 영상의 인덱스는 움직임 벡터와는 달리, 8x8단위의 블록으로 최대 전송할 수 있다. 물론, 매크로블록이 네 개 이하의 블록으로 전송될 때는 파티션 블록의 개수 만큼만 시그널링된다.

3) 움직임 벡터와 보간방법

앞서 기술한 것처럼 각각의 파티션 블록별로 움직임 벡터가 필요하다. 각각의 움직임 벡터의 값은 이웃하는 블록의 움직임 벡터에 기반해서, 움직임



〈그림 4〉 현재 블록의 이웃하는 블록의 정의

벡터를 추정하고, 그 추정된 값과 실제 움직임 벡터의 값의 차분값 만이 디코더단으로 전달된다. 움직임 벡터를 추정하는 방법은 기본적으로, 왼쪽의 블록, 위쪽의 블록, 오른쪽 위의 블록의 움직임 벡터의 중간값(median 값)을 추정치로 한다. 하지만, 각 매크로블록이 어떤 형태로 나뉘어졌는가에 따라서 예외적인 경우도 존재하며, 자세한 내용은 스펙문서에 기술되어 있다.

참고로, 이웃하는 블록을 결정하는 방법은 〈그림 4〉에 보였다. 현재 블록에서 왼쪽에 위치한 'a' 픽셀을 포함하는 블록이 왼쪽의 블록이 되며, 'b' 픽셀을 포함하는 블록이 위쪽의 블록, 'c' 픽셀을 포함하는 블록이 오른쪽 위의 블록이 된다. 만일, 'b'와 'c' 픽셀이 하나의 블록 혹은 파티션에 포함되어 있다면, 위쪽 블록과 오른쪽 위의 블록은 같은 블록이 된다.

움직임 벡터가 가질 수 있는 값은 H.264의 Level에 따라서 달라지며, 움직임 벡터는 luma 픽셀의 경우 1/4 픽셀까지의 해상도를 가질 수 있다. Chroma성분의 움직임 벡터는 그 블록에 해당하는 luma의 움직임 벡터를 그대로 사용한다.

움직임 벡터가 1/4 픽셀의 해상도를 가지므로, 정수 픽셀의 참고 영상으로부터 해당하는 해상도를 구하기 위해서는 보간법을 사용해야 한다. H.264는

세 가지의 보간법을 사용한다. 첫 째, luma영상의 1/2 해상도를 위해서는 [1 -5 20 20 -5 1]의 FIR 필터를 사용하고, 둘째, 1/4 해상도를 위해서는 1/2 해상도로 보간된 픽셀 값과 정수 해상도의 픽셀 값을 이용해서 보간한다. 마지막으로, chroma의 경우는 정수 해상도의 픽셀 두 값을 선형 보간법으로 1/8 해상도의 픽셀값을 구한다.

보간법에 필요한 픽셀의 위치가 참고 영상의 외부에 위치할 경우에는 참고하는 영상의 경계에 존재하는 픽셀의 값을 이용해서 보간에 사용한다.

3. 양자화와 변환 방법

영상을 엔코딩하는 과정에서는, 영상이 가지는 픽셀값과 양 절에서 기술한 알고리즘으로 예측한 값의 차분을 변환하고, 그 값을 1차원의 배열로 만든다. 1차원의 배열값을 양자화하고, 양자화한 값을 엔트로피 코딩을 통해 압축한 형태로 디코더쪽으로 전송한다.

블록에 기반한 기존의 압축 알고리즘에서는 DCT에 기초하였으나, H.264에서는 integer 변환을 통하여 정확성의 손실없이 변환이 가능하게끔 하였으며, 연산의 중간값의 범위도 16비트 연산으로 한정함으로써 하드웨어로 구현함에 있어서 효율성을 얻을 수 있게끔 하였다. 대부분의 변환은 4x4 integer 변환을 사용하였고, DC 값을 변환하는 과정에서는 Hadamard변환을 사용한다. 또 하나의 변환과 관련한 특징으로는 Intra_16x16의 형태의 매크로블록에 대해서는 차분값의 변환계수중 DC값의 변환계수들만을 한번 더 변환 과정을 거치게끔 한다. 이를 통해서 0이 아닌 변환계수를 최소화하는 효과를 얻을 수 있다.

H.264에서는 양자화를 52단계로 나누어서 사용

한다. QP값이 6씩 증가하면, 양자화의 단위가 두 배씩 증가하게끔 로그스케일로 양자화가 설계되어 있다. 양자화에 필요한 QP값은 픽처단위에서 초기 값을 설정해 줄 수 있고, 슬라이스 단위, 혹은 매크로블록 단위까지 차분값을 이용할 수 있으며, 이는 각 매크로블록 별로 다른 양자화 단계를 이용할 수 있다는 의미이다. Chroma성분에서 사용하는 양자화 단계는 luma성분에서 사용하는 양자화의 단계 보다 작은 단계를 사용한다.

양자화한 변환 계수들은 두 가지 방법으로 1차원 배열로 만드는데, 지그재그 방법과 필드 방법이 있다. 필드 방법은 비월 주사방식의 매크로블록에 해당하는 변환 계수들에 해당하며, 그렇지 않을 경우에는 지그재그 스캔의 방법으로 1차원 배열을 만든다.

4. 엔트로피 코딩 방법

엔트로피 코딩방법은 주어진 데이터에서 나타나는 통계적 특성을 이용하여, 효율적으로 압축하는 기법을 말한다. H.264에서 사용하는 엔트로피 방법은 크게 세가지로 나눌 수 있다. 하나는 universal coding방법의 일종인 Exp-Golomb코딩 방법과, 컨텍스트를 이용한 가변 길이 코딩방법(CAVLC), 그리고 컨텍스트를 이용한 산술 코딩방법(CAVC)이다. 이 중에서 컨텍스트를 이용한 산술 코딩방법은 참고문헌으로 남기고, 여기서는 앞의 두 가지 방법에 대해서 설명하기로 한다.

1) Exp-Golomb 코딩 방법

모든 신택스 요소들에 대해서 무한 길이의 코드 워드를 하나의 방법을 나타내는 방법의 일종으로, 표 1과 같이 코드의 형태와 그 코드가 나타내는 값을 나타냈다.

〈표 1〉 Exp-Golomb코딩으로 나타낼 수 있는 영역의 예

비트열의 형태	코드값이 가지는 영역
1	0
0 1 x_0	1-2
0 0 1 x_1, x_0	3-6
0 0 0 1 x_2, x_1, x_0	7-14
0 0 0 0 1 x_3, x_2, x_1, x_0	15-30
0 0 0 0 0 1 x_4, x_3, x_2, x_1, x_0	31-62

코드가 나타내는 값들을 다음과 같이 구분할 수 있다.



- 부호를 가지는 값: 매크로 블록 타입(mb_type), 슬라이스 타입(slice_type), 매크로블록을 추가정보 없이 전송할 개수(mb_skip_run) 등
- 부호를 가지는 값: 움직임 벡터의 차분의 크기(mvd_l0, mvd_l1), 양자화 크기의 차분(slice_qp_delta, pic_init_qp_minus26) 등
- 조건에 따라 달라지는 값: 어떤 픽처를 참고할 것인가에 대한 인덱스(ref_idx_l0, ref_idx_l1)
- 코드를 대응시켜야 하는 값들: 블록이 어떻게 코딩되어 있는 가에 대한 값(coded_block_pattern)

2) 컨텍스트를 이용한 가변길이 코딩방법(CAVLC)

컨텍스트라 함은 현재 코딩 혹은 디코딩하고 있는 주변의 값들을 일컫는다. 이런 컨텍스트를 이용함으로써 보다 효율적으로 압축을 할 수 있다. H.264에서는 변환 계수를 코딩하는데 CAVLC를 사용하고 있다. 양자화한 계수를 지그재그 스캔하고, 1차원으로 만든 배열에서 변환계수가 '0' 인 개수와 '0' 이 아닌 변환 계수의 절대값, 그리고, 4x4블록에서 마지막으로 이어지는 '1'의 개수(trailing one's)를 코딩하는데 이 방법이 사용되고 있다. 주변 블록이 가지는 '0' 이 아닌 계수의 개수로 컨텍스트를 정하고, 이 컨텍스트에 따라 다섯 개의 코드 테이블을

정의한다. 각각의 테이블에서는 하나의 블록에 '0'이 아닌 계수의 개수와 마지막으로 이어지는 '1'의 개수를 묶어서 코드화 시켰다.

5. 디블록킹 과정(Deblocking process)

변환영역에서 정보량을 줄이기 위해서 사용하는 양자화는 정보량의 축소라는 긍정적인 측면이 있는 반면, 복원된 영상에서 각 블록의 경계에서 블로킹 효과가 생기는 부정적인 측면도 야기시킨다. 이전의 압축 알고리즘에 있어서는 그러한 블로킹 효과를 제거하기 위해서 후처리 필터를 디코더 단에서 선택적으로 사용할 수 있으며, 기타 다른 화질 개선의 알고리즘과 더불어 사용한다. 그에 반해서, H.264의 경우에는 블로킹 효과를 줄여주는 필터를 디코딩 루프안에 둬으로써, 움직임관련 예측에서 보다 나은 성능을 발휘할 수 있게끔 하였다.

비록 디코딩 루프안에 필터를 위치시켰지만, 디코딩 과정과 디블록킹 과정을 완전히 분리하고 있다. 현재 디코딩하는 매크로블록이 참고해야 하는 주변 픽셀 값 들은 디블록킹 과정을 거치기 전의 값을 참조하게끔 정의되어 있다.

디블록킹 과정은 크게 두 가지로 나눌 수 있다. 하나는 경계의 세기를 결정하고, 또 하나는 경계의 세기에 따라서 실제로 픽셀값을 필터링하는 과정이다.

블록간 경계의 세기를 결정하기 위해서는, 매크로블록의 타입, 블록간의 경계가 매크로블록의 경계여부, 각 블록이 변환영역에서 '0'이 아닌 변환계수를 가졌는지 여부, 움직임 벡터의 크기, 참조하는 픽처의 동일성 여부를 사용하여 결정한다. 이러한 매개변수들을 디코딩 과정에서 사용하고 바로 디블록킹 과정에서 사용할 수 있다면, 매크로블록 단위의 파이프라인 형태의 구현이 쉽다. 하지만, 앞서 말

한 바와 같이, 디블록킹 과정과 디코딩 과정이 분리되어 있기 때문에 이러한 매개변수의 관리를 효율적으로 관리하고 이용하기 위해서는 세심한 설계가 필요하다. 전적으로 소프트웨어로 구현하는 경우에는 큰 어려움이 없으나, 하드웨어 가속기를 설계하기 위해서는 이와 같은 매개변수의 관리가 하드웨어의 성능을 결정하는 요인이 된다.

매개변수를 이용한 경계의 세기를 결정하면, 그 세기에 따라서 필터링한다. 필터링에 관한 상술은 생략하도록 한다.

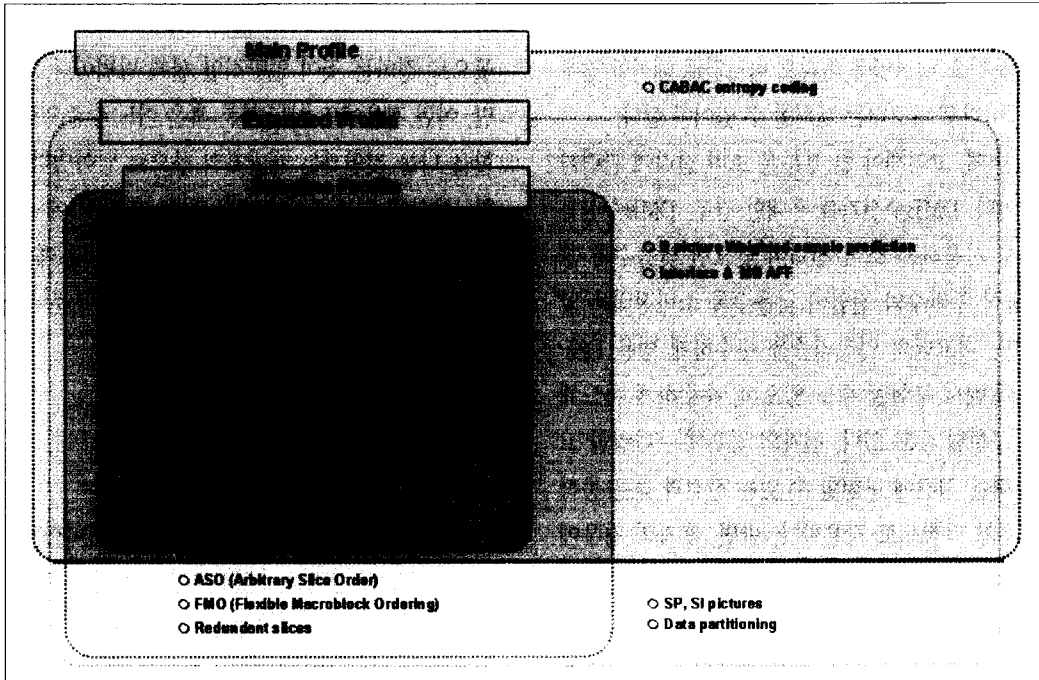
디블록킹 과정에서 chroma성분의 필터링은 4x4블록을 기준으로 이루어지는데, chroma성분의 경계에 해당하는 luma성분의 경계의 세기를 이용해서 필터링한다. 움직임 보상의 경우, chroma 성분은 해당하는 luma성분의 움직임 벡터에 따라서 2x2크기까지 움직임 보상을 수행하는 반면, 디블록킹 과정에서는 최소 단위가 4x4블록 단위로 하게끔 정의하고 있다.

IV. DMB에서의 H.264

1. 프로파일 및 레벨

H.264에서 정의하는 프로파일을 High profile을 제외하고 <그림 5>에서 나타내었다. 이 중에서 DMB는 가장 기본적인 baseline 프로파일에서 안쪽 검은 실선으로 표시한 영역만을 사용한다. Eureka-147에 기반한 DMB에서 보다 낮은 에러율을 보장하기 위해서 추가적으로 외부호화 과정을 거침으로써, error환경에 대비한 몇 가지의 도구는 제외시켰다.

DMB에서 사용하는 레벨은 기본적으로 H.264에서 정의하는 1.3 레벨을 따랐으며, 방송환경에 적합



〈그림 5〉 H.264이 정의하는 프로파일(High profile제외)

하도록 몇 가지의 제약사항을 더 두었다. 지원하는 포맷은 QCIF, QVGA, CIF와 WDF로 정의했으며, WDF(가로 384, 세로 224픽셀)는 향후 지상파 방송이 디지털 전환후 대부분의 콘텐츠가 16:9형태로 제작될 예정에 있으며, 이런 콘텐츠를 DMB로 재전송할 경우를 대비해서 포함시켰다. H.264는 참고영상의 수를 매크로블록의 개수로 정의하고 있지만, 국내 지상파 DMB의 경우, P 매크로블록이 참고할 수 있는 참고영상의 수를 방송이 사용하는 포맷의 세 장으로 제한하고 있다.

2. 구현과정에서 고려해야 할 점

DMB는 이동환경에서 수신을 전제로 한다. 이 전제로 인해서 개발단계에서 몇 가지 고려해야 할 점

이 있다. 그 중에서 가장 중요한 두 가지를 언급하도록 한다.

먼저, 여러환경에서의 성능이다. 이동 환경에서의 비디오 수신이기 때문에, Eureka-147에서 정의하는 채널코딩 기법 이외에 추가적으로 외부호화 과정을 거친다. 그럼에도 불구하고 언제나 충분한 전파의 수신을 보장할 수 없다. 따라서, 전송되는 비트열에서 에러를 피할 수 없으며, 이러한 에러가 발생했을 때에도 비디오 디코더가 강인하게 대처를 해야 한다. 받은 비트열에서 오류가 발견되면, 그 위치에 따라서, 오류를 무시하고 넘어갈 수 있는 부분도 있고 혹은 다음 매크로블록, 다음 슬라이스, 다음의 픽처로 건너뛰어야 하는 경우도 있다. 비디오 비트 스트림에서 오류가 발생하지 않고, 전송단(Transport 단)에서 오류가 발생할 경우에는 때에 따라 치명적

인 오류를 발생시킬 수도 있다. 예를 들어, 에러 때문에 오디오 비트열을 비디오 비트열로 해석할 경우라도, 비디오 디코더는 대처할 수 있어야 한다.

두 번째, 고려해야 할 부분은 소비 전력에 관한 문제이다. DMB수신기를 휴대할 경우, DMB수신을 위해서 사용하는 소비 전력이 낮으면 낮을수록 좋다. 동작 주파수와 전압이 낮을수록 소비전력은 낮아진다. 디코더의 알고리즘을 순수하게 범용 DSP를 사용해서 구현할 수도 있으며, 순수하게 하드웨어로 구현할 수도 있다. 전자의 경우에는 다양한 코덱을 빠른 시간에 구현할 수 있는 장점과 높은 동작 주파수가 단점으로 작용한다. 반면, 후자의 경우에는 설계 및 검증, 오류에 대한 대처가 힘든 단점과 낮은 동작주파수로 소비전력을 절약할 수 있는 장점이 있다. 어떤 해법을 선택할 것인가는 시장의 요구사항에 따라서 달라진다. 어느 수준의 소비전력까지 시장에서 받아들여질 수 있을까 하는 문제에도달한다. 이런 문제를 풀기 위해서 근래에는 하드웨어와 소프트웨어를 동시에 사용할 수 있는 플랫폼들이 많이 개발되어 있다. 순서가 정해져 있고, 계산량이 많고 발생 빈도가 높은 부분은 하드웨어의 몫으로 하고, 순서가 유동적이며 알고리즘이 복잡

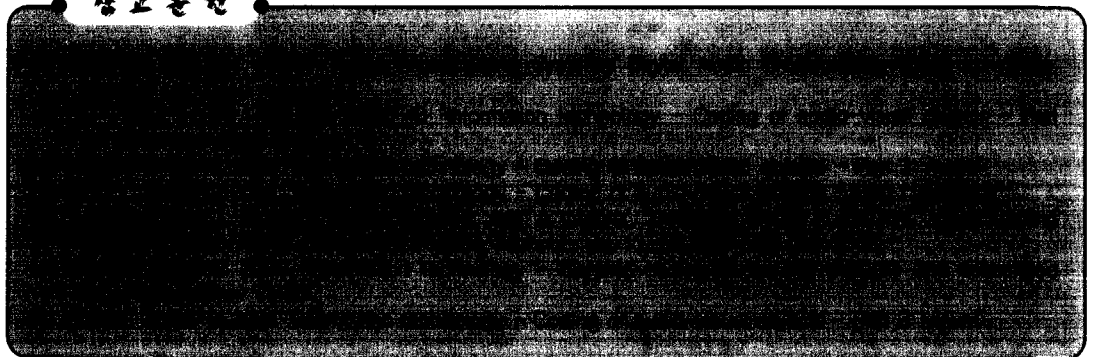
하거나, 자주 일어나지 않는 부분은 소프트웨어의 몫으로 정하는 것이 일반적인 설계 방법이다. 하지만, 이런 경우에도 어느 부분 혹은 어느 수준으로 역할을 나눌 것인가는 개발자의 몫으로 남아있다. 또한, 동시에 하드웨어와 소프트웨어를 동시에 작동하기 위해서는 각각 블록간의 프로토콜 및 예외상황의 처리 등은 항상 염두에 두어야 하는 부분이다.

V. 맺음말

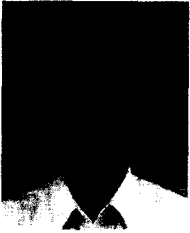
새로운 방송환경으로 자리잡을 DMB서비스가 사용할 H.264를 간략하게 살펴보았다. 낮은 비트율에서도 기존의 코딩 기법보다는 좋은 화질을 제공하기 위해서 여러가지 기법들을 사용하였다.

DMB는 방송환경에서 H.264가 최초로 상용화되는 서비스이다. DTV에 사용하는 MPEG2 비디오의 경우처럼, 방송의 초기의 영상화질보다 기술의 발전에 따라서 엔코딩 기법이 발전하면서 꾸준히 화질의 개선이 일어날 것으로 예상된다. 또한, 디코더의 구현에 관한 알고리즘을 개선함으로써, 구현에 필요한 소모 전력을 절약하는 방법도 꾸준히 이루어질 것이다.

참고문헌



필자 소개



이철수

- 1996년 : 부산대학교 전자공학과 학사
- 1998년, 2002년 : 한국과학기술원 전기 및 전자공학과 석사 및 박사
- 2004년 현재 : LG전자 DTV연구소 선임연구원(DMB SoC개발 참여)