

# 모바일 단말기 기반에서 H.264/AVC 디코더의 디블록킹 필터(De-Blocking Filter) 구현 방안\*

김송주\*, 김대곤\*, 유철중\*, 장옥배\*  
\*전북대학교

e-mail:{songju, bluve0214, cjyoo, okjang}@chonbuk.ac.kr

## De-Blocking Filter Of H.264/AVC Decoder Implementation Based on Mobile Device

Song-Ju Kim\*, Dae-Gon Kim\*, Cheol-Jung Yoo\*, Ok-Bae Chang\*  
\*Chonbuk National University

### 요 약

국제 비디오 압축 표준인 H.264/AVC는 MPEG-2나 MPEG-4등에 비해 압축률이 2배 이상 향상되어 저대역폭을 가지는 모바일 단말기 기반의 네트워크에서도 이전의 압축표준보다 훨씬 좋은 품질의 영상을 제공하나 높아진 압축률에 상응하여 복잡도 또한 증가하였다. 이러한 복잡도를 해결하기 위하여 디코딩을 하는 과정에서 병목현상을 일으키는 부분들을 하드웨어의 최적화된 설계로 해결해왔다. 이러한 하드웨어 기반 해결은 단말기의 교체라는 단점을 가지고 있다. 이러한 단점을 해결하기 위하여 본 논문에서는 소프트웨어 디코더가 모바일 단말기에 적용되기 위한 조건들을 살펴보고 디코딩 과정 중에 가장 많은 병목 현상을 가지는 디블록킹 필터(De-Blocking Filter)를 모바일 단말기에 적용하기 위한 방법을 제안한다. 이러한 시도는 모바일 단말기 상에서 하드웨어 기반 디코더가 아닌 소프트웨어 기반 디코더가 구현 될 수 있는 기초가 된다.

### 1. 서 론

H.264/AVC는 ITU-T H.264와 ISO/IEC 14496-10 (MPEG-4 Part 10)으로 승인된 새로운 국제 비디오 압축 표준이다[1]. H.264/AVC는 기존의 압축 표준들에 비하여 월등한 압축 성능으로 다양한 멀티미디어 서비스를 제공하고 있다.

현재 우리나라에서는 H.264/AVC가 위성 방송 서비스인 S-DMB, T-DMB 표준 비디오 디코더로 지정되어 이동 통신 산업에서 멀티미디어 기술의 핵심기술로 인정되고 있다. 일부 이동통신사에서는 H.264/AVC를 통하여 스트리밍 서비스를 제공하고 있는 등 이동통신 산업에서 빼놓을 수 없는 중요한 기술이 되었다.

디지털 동영상 통신 시스템이 모바일 단말 통신 시스템 상에서 실현되기 위해서는 유선 네트워크에 비하여

무선 네트워크가 상대적으로 열악하다는 점을 감안하여 모바일 단말의 전력과 메모리의 성능을 고려한 낮은 복잡도를 지닌 디코딩 방법을 필요로 하고 있다.

또한, 모바일 단말기에 하드웨어 칩 형태로 적용되기 시작하였으나 하드웨어 기반의 디코더는 신기술을 적용하기 위해서는 모바일 단말기의 교체라는 단점을 가지고 있다. 하드웨어적 구현이 아닌 소프트웨어적 구현의 필요성이 제기되어 모바일 단말기에 적용 가능한 소프트웨어 디코더의 개발이 필요하게 되었다.

H.264/AVC 소프트웨어 디코더는 현재까지도 꾸준히 성능 향상을 보이고 있으며 보다 나은 성능을 위해 계속해서 추가적인 표준이 발표되고 있으나, 이러한 소프트웨어 디코더는 모바일 단말기에서 수행되는 것을 고려하고 있지는 않다.

모바일 단말기는 그 특성상 넓은 영역의 디스플레이 크기를 가지고 있지 않으며, 저전력 구조와 저성능의

\* 본 연구는 한국과학재단 「기초연구지원사업(특정기초)-과제번호 R01-2004-000-10730-0」에 의해 지원되었습니다.

CPU구조를 지니고 있다. 따라서 모바일 단말기 특성을 기반으로 하여 디코더를 통한 화질 저하를 어느 정도 감수할 수 있으며, 이러한 화질 저하를 통해 복잡도 및 전력을 적게 소모하는 형태의 H.264/AVC 소프트웨어 디코더를 설계할 수 있다.

H.264/AVC 소프트웨어 디코더가 인코딩된 영상을 디코딩함에 있어 가장 많은 병목 현상을 보이는 부분 중에 하나가 디블록킹 필터(De-Blocking Filter) 적용에 관한 부분이다.

디코더 수행에 있어서 디블록킹 필터(De-Blocking Filter)가 적용되는 알고리즘을 살펴보고 이것을 모바일 단말기에 적용시킬 수 있는 방안을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 H.264/AVC 소프트웨어 디코더 알고리즘 중 디블록킹 필터(De-Blocking Filter)에 대하여 알아보고 3장에서는 디블록킹 필터(De-Blocking Filter)를 모바일 단말기에 적용하기 위한 방법을 제안한다. 4장에서는 제안한 방법 형식으로 참조 소프트웨어를 수정하여 인텔 기반의 컴퓨터에서 테스트한 결과에 대하여 알아보고 마지막으로 5장에서는 결론 및 향후 연구에 관하여 기술한다.

## 2. 관련 연구

H.264/AVC 디코더는 블록 기반의 압축 방식을 사용하고 있으며, 이러한 블록 기반의 압축 방식은 저비트율로 압축될 경우 블록간의 경계가 뚜렷하게 보이는 블로킹 현상(Blocking Artifacts)이 나타난다. 이러한 블로킹 현상을 제거하기 위하여 H.264/AVC 디코더에서는 루프 필터 방식의 디블록킹 필터(De-Blocking Filter)를 사용하고 있다[2]. 이 필터의 연산량은 전체 복호기 연산량 중 많은 부분을 차지하고 있으므로 필터를 구성하고 있는 각 요소에 대하여 살펴보고 디블록킹 필터(De-Blocking Filter)를 모바일 단말기에 맞게 소프트웨어 기반의 디코더로 구현하기 위한 방법을 살펴본다.

### 2.1 디블록킹 필터(De-Blocking Filter) 알고리즘

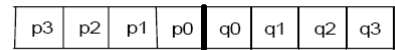
디블로킹 필터(De-Blocking Filter)를 이용하여 영상의 블로킹 현상을 제거하여 양호한 복호화된 영상을 얻게 되지만 이 경우 전체 처리량의 50%를 점유하는 경우도 있다[3]. 이를 해결하기 위해 H.264/AVC 표준에서는 비트열 중 픽처 파라미터 세트(Picture Parameter Set)에 포함되는 `deblocking_filter_control_present_flag`와 슬라이스 헤더에 포함되는 `disable_deblocking_filter_idc`라는

두개의 파라미터에 의해

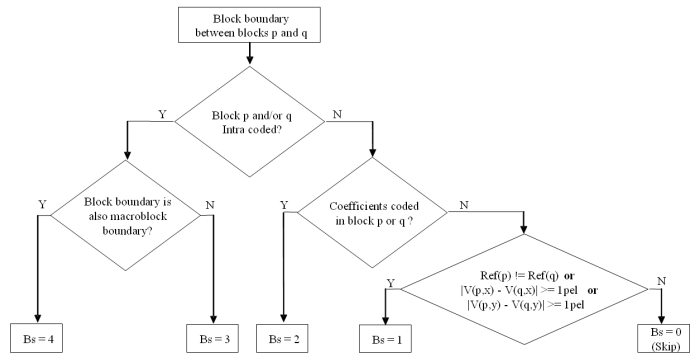
- (1) 블록 경계와 매크로 블록 경계에 필터를 적용
- (2) 매크로 블록 경계에만 필터를 적용
- (3) 필터를 사용하지 않음

의 세가지 방법 중 한 가지를 사용자가 선택하도록 하고 있다.

이러한 파라미터에 의해 필터 적용을 하기에 앞서 블록 경계 강도( $bS$  : boundary Strength) 값을 결정해야 한다. 블록 경계 강도는 [그림1]에서와 같이  $p_0 \sim p_3$ 로 구성된 블록과  $q_0 \sim q_3$ 로 구성된  $16 \times 16$  휘도(luma) 블록과  $8 \times 8$  색차(chroma) 블록에 대해서 [그림2]와 같은 결정을 통해  $\{0, 1, 2, 3, 4\}$ 의 값을 가진다.



[그림1] 경계 필터링을 위한 참조 픽셀



[그림2] bS 결정조건

즉, 동일한 프레임의 블록경계에 모두 같은 강도의 필터링을 적용하지 않고, 블록 경계가 매크로블록 경계인지 아닌지, 또는 어느 블록이 화면 내 부호화를 사용하는지의 여부에 따라 어느 정도의 강도로 필터링을 적용할 것인지를 결정한다.

이렇게 적응적으로 필터링을 함으로써 블록왜곡이 생기기 쉬운 경계에 강한 필터링에 의해 왜곡을 제거하며, 블록왜곡이 생기기 힘든 곳에서는 불필요한 필터링 처리 때문에 연산량이 증가하는 것을 막는다.

블록 경계 강도( $bS$ ) 값이 규정되어지면, 즉 경계선이 나타내는 조건에 따라  $bS$  값이 0과 4사이에서 정해지면 다음 조건이 모두 만족하는 경우에 필터링을 적용하게 된다.

- (1)  $bS > 0$
- (2)  $|p_0 - q_0| < \alpha$

- (3)  $|p_1 - p_0| < \beta$
- (4)  $|q_1 - q_0| \leq \beta$

여기에서  $\alpha$ ,  $\beta$  는 H.264/AVC 표준에서 상수로 정의되어 있다. [그림3]은 표준 문서에 나와 있는 값이다.

Table 8-14 – Derivation of indexA and indexB from offset dependent threshold variables  $\alpha$  and  $\beta$

		indexA (for $\alpha$ ) or indexB (for $\beta$ )																										
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
$\alpha$		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	5	6	7	8	9	10	12	13
$\beta$		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	3	3	3	3	4	4	4	4

Table 8-14 (concluded) – Derivation of indexA and indexB from offset dependent threshold variables  $\alpha$  and  $\beta$

		indexA (for $\alpha$ ) or indexB (for $\beta$ )																									
		26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
$\alpha$		15	17	20	22	25	28	32	36	40	45	50	56	63	71	80	90	101	113	127	144	162	182	203	226	255	255
$\beta$		6	6	7	7	8	8	9	9	10	10	11	11	12	12	13	13	14	14	15	15	16	16	17	17	18	18

[그림3]  $\alpha$ ,  $\beta$  정의

모든 조건이 만족하는 경우에는 블록 경계 강도(bS)의 값에 따라 각기 다른 조건으로 필터가 사용된다.

[표1]은 디블록킹 필터(De-Blocking Filter)의 적용시 bS=4 인경우의 의사 코드이다[4].

[표 1] bS=4 인 경우 필터 적용 의사코드

```
Alpha = ALPHA_TABLE[indexA];
Beta = BETA_TABLE[indexB];
C0 = ClipTable[indexA][bS]

• Filtering of edges with bS=4

if(|p2-p0| < Beta && |p0-q0| round(Alpha/4))
    P0 = (p2+2p1+2p0+2q0+q1)/8 // 5-tap
    P1 = (p2+p1+p0+q0)/4 // 4-tap
    P2 = (2p3+3p2+p1+p0+q0)/8 // luma only
else
    P0 = (2p1+p0+q1)/4

if(|q2-q0| < Beta && |q0-q0| round(Alpha/4))
    Q0 = (q2+2q1+2q0+2p0+p1)/8 // 5-tap
    Q1 = (q2+q1+p0+p0)/4 // 4-tap
    Q2 = (2q3+3q2+q1+q0+p0)/8 // luma only
else
    Q0 = (2q1+q0+p1)/4
```

H.264/AVC 표준에서 제안하는 디블록킹(De-Blocking Filter) 알고리즘 전체를 모바일 단말기에 적용하기에는 블록 경계 강도 값을 결정해야하는 부분과 각 블록 경계

강도 값에 따라 각기 다른 FIR 탭 필터를 모두 적용해야 하므로 많은 연산량이 필요하다.

### 3. 모바일 단말기를 위한 디블록킹 필터

#### 3.1 블록 경계 강도 선택 알고리즘 변경

모바일 단말기를 위해 블록 경계 강도(bS) 결정 알고리즘을 변경하는 것에 대한 전제 조건은 인간의 시각 시스템은 휘도에 비해 채도에 덜 민감하다는 것이다.

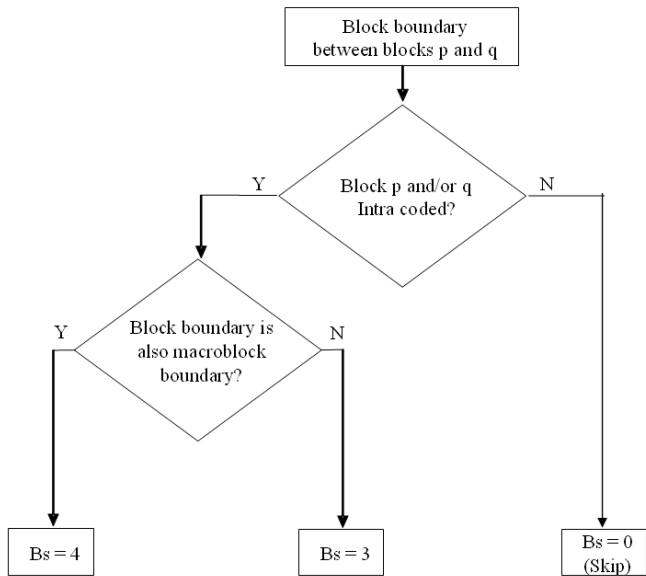
이를 이용하면 매크로 블록 단위로 블록 경계 강도 값을 구할 때에 비하여 훨씬 더욱 연산량을 줄일 수 있다. 즉 매크로 블록 단위로 하는 경우엔 휘도 블록뿐만 아니라 채도 블록 까지도 디블록킹 필터(De-Blocking Filter)를 적용해야 하지만 휘도 블록에 대해서만 디블록킹 필터(De-Blocking Filter)를 거치도록 수정하게 되면 상당한 양의 연산량을 줄일 수 있다. 따라서 휘도 블록에 대해서만 디블록킹(De-Blocking Filter)를 적용하는 방안으로 알고리즘을 변경하는 것으로 제안한다.

#### 3.2 블록 경계 강도에 따른 필터링 적용 제한

필터링 적용을 제한하는 전제 조건은 모바일 단말기의 디스플레이 영역의 크기가 제한되어 있다는 점이다.

디스플레이 영역의 크기가 제한되어 있으므로 큰 영역에서 보여주는 영상보다 블록킹 현상에 덜 민감해진다. 따라서, 본 논문에서는 필터링 되어지는 값을 bS=4, 3 인 경우만으로 제한하고자 한다. 즉, 인트라 코드 되어 있고 블록 경계인 경우에 가장 큰 필터링을 실시하고 블록경계가 아닌 경우에는 한 단계 낮은 필터링을 한다. 화면이 인트라 코딩 되어 있지 않다면 필터링을 하지 않는다. 따라서 필터링 적용이 제한되어지므로 그만큼 연산량은 줄어들게 된다.

제안 방법을 도식화하면 [그림5]과 같다.



[그림5] 필터링 적용수 제한

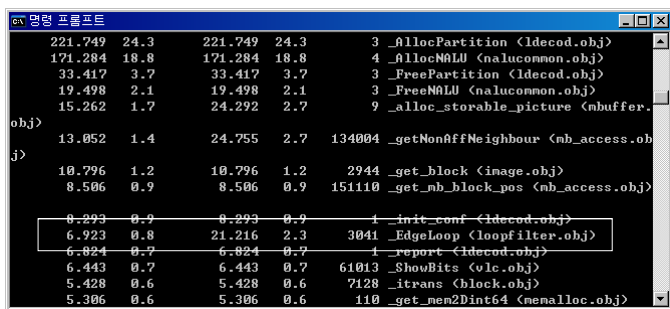
4. 실험결과

알고리즘 변경 전과 후를 VC++를 이용하여 Profile 해 본 결과는 [표3]과 같다.

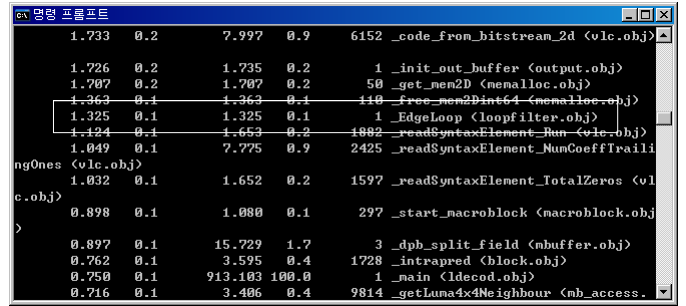
[표3] 적용 전 후 디코더 프로파일

profile	Function Time (ms)	Function
적용전	6.923	EdgeLoop
적용후	1.143	EdgeLoop

알고리즘 변경을 통하여 약 7배 정도 나은 성능 향상을 보임을 알 수 있다. 이 결과에 대한 캡처 결과는 [그림 6],[그림 7]과 같다.



[그림 6] 적용 전 프로파일



[그림 7] 적용 후 프로파일

5. 결론 및 향후 연구

H.264/AVC는 새로운 압축표준으로 기존의 압축표준들과 비교하여 압축률도 향상되었고 좋은 화질도 제공한다, 하지만 압축률과 화질 향상 두 가지 모두를 이루기 위해 연산량이 많이 증가하여 모바일 단말기에서 소프트웨어 기반으로 디코더를 만들기에 더욱 힘들어졌다.

현재에는 H.264/AVC기반 디코더는 소프트웨어 기반으로 하기에는 너무 많은 연산량과 전력 소비가 커서 하드웨어 기반으로 구현되고 있는 실정이다. 하지만 본 논문에서 소프트웨어 디코더의 성능을 향상시켜보고 또한 모바일 단말기의 성능도 나날이 발전하고 있음에 주목하여 볼 때 H.264/AVC의 소프트웨어 디코더 구현 또한 성공 가능성이 낮은 것만은 아님을 알게 되었다.

이번 연구에서는 소프트웨어 디코더의 성능 향상을 위해 인텔 CPU 기반에서 알고리즘을 변경하고 테스트 하였다. 이러한 결과로 디코더의 성능 향상에는 많은 증가를 가져왔지만 영상의 크기가 커지는 경우에는 PSNR 값이 무척 떨어지는 결과를 가져왔다. 이유는 블록 경계 강도 값을 결정할 때 변경한 알고리즘 때문임이 밝혀졌다. 그러나 본 논문에서 지향하는 것은 모바일 단말기 상에서 소프트웨어 디코더를 통해 영상을 복호화 하는 것에 중점을 두고 있어 실 사용자가 보기에는 많은 저하를 가져오지 않았다. 향후에는 실제 ARM 코어 기반에서 변경된 알고리즘을 적용하여 모바일 단말기에 최적화된 소프트웨어 디코더를 연구하는 것이 필요하다.

참 고 문 헌

[1] ISO/IEC ITU-T Rec. H.264, Advanced Video Coding, 2003.  
 [2] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M.Karczewicz, "Adaptive deblocking filter", IEEE Trans. Circuits and Systems for Video Technology,

Vol. 13, no. 7, pp. 614-619, Jul. 2003.

[3] Y. Huang, T. Chen, B. Hsieh, T.Wnag, T.Chang, and L. Chen, "Architecture design for deblocking filter in H.264/JVT/AVC", Proceedings of International Conference on Multimedia and Expo, pp. 693-696, 2003.

[4] H.264/AVC Reference Software Version JM10.2, <http://bs.hhi.de/~suehring/tml/>, March 2003