

논문 2010-47SP-3-4

저 복잡도 비디오 복호화기를 위한 디블록킹 필터

(Deblocking Filter for Low-complexity Video Decoder)

조현호*, 남정학*, 정광수*, 심동규**, 조대성***, 최웅일***

(Hyun-Ho Jo, Jung-Hak Nam, Kwang-Su Jung, Dong-Gyu Sim, Dae-Sung Cho, and Woong-Il Choi)

요약

본 논문은 저 복잡도 비디오 복호화기를 위한 디블록킹 필터를 제안한다. 휴대전화와 같은 모바일 장치에서 많이 사용되는 H.264/AVC Baseline 프로파일은 MPEG-4 Visual 보다 압축 성능은 두 배 이상 높지만, 1/4-픽셀 보간 필터, 적응적 엔트로피 모델 및 디블록킹 필터를 사용함에 따라 복호화기의 복잡도가 높다는 문제점이 있다. 본 논문에서는 H.264/AVC의 부호화 성능은 유지하면서 복호화기의 복잡도를 감소시키기 위하여 저 복잡도 디블록킹 필터를 제안한다. 본 논문에서 제안된 저 복잡도 디블록킹 필터는 BS (Boundary Strength)값에 대해 CBP (Coded Block Pattern)값을 이용하여 계산함으로써, 기존의 방법보다 분기문의 수를 49% 감소시켰다. 또한, 인트라 매크로블록 경계에서 적용되는 강한 필터링 (Strong Filtering)에 대해 필터링의 적용 범위를 두 픽셀로 제한하였다. 실험 결과, 제안하는 저 복잡도 디블록킹 필터는 H.264/AVC Baseline 프로파일에 비해 BDBitrate를 -0.02% 감소 시켰고, 디블록킹 필터의 복잡도는 42%, 복호화기 전체의 복잡도는 8.96% 감소 시켰다.

Abstract

This paper presents deblocking filter for low-complexity video decoder. Baseline profile of the H.264/AVC used for mobile devices such as mobile phones has two times higher compression performance than the MPEG-4 Visual but it has a problem of serious complexity as using 1/4-pel interpolation filter, adaptive entropy model and deblocking filter. This paper presents low-complexity deblocking filter for decreasing complexity of decoder with preserving the coding efficiency of the H.264/AVC. In this paper, the proposed low-complexity deblocking filter decreased 49% of branch instruction than conventional approach as calculating value of BS by using the CBP. In addition, a range of filtering of strong filter applied in intra macroblock boundaries was limited to two pixels. According to the experimental results, the proposed low-complexity deblocking filter decreased -0.02% of the BDBitrate comparison with baseline profile of the H.264/AVC, decreased 42% of the complexity of deblocking filter, and decreased 8.96% of the complexity of decoder.

Keywords : H.264/AVC, 디블록킹 필터, 저 복잡도 비디오 복호화기, Boundary Strength, CBP

I. 서론

디지털 비디오를 압축하는 기술인 비디오 코덱은 여

러 표준화 단체, 회사, 연구소들에 의해 개발되고 있다. 이러한 비디오 코덱들 중 H.261, H.262, MPEG-1, MPEG-2, MPEG-4, H.263+등의 기술은 ITU-T의 VCEG (Video Coding Expert Group)과 ISO/IEC의 MPEG (Moving Picture Expert Group)에 의해 개발되어 국제 표준들로 제정되었으며, 현재 개인용 컴퓨터부터 여러 전자 기기들에 이르는 다양한 분야에서 널리 사용되고 있다.

2003년 5월에는 VCEG과 MPEG이 공동으로 참여하는 JVT (Joint Video Team)에 의해 H.264/AVC라는 고효율의 비디오 코덱의 표준화가 완료되었고, 이는 현존하는 비디오 코덱 중 가장 부호화 효율이 뛰어난 것

* 학생회원, ** 정회원, 광운대학교 컴퓨터공학과
(Dept. of Computer Engineering, Kwangwoon University)

*** 정회원, 삼성전자(주)
(Samsung Electronics)

※ 본 연구 일부는 삼성전자(주)의 지원을 통하여 이루어졌음

※ 본 연구 일부는 서울시 산학협력사업(10570)으로 구축된 서울 미래형콘텐츠컨버전스 클러스터 지원으로 수행되었음

접수일자: 2009년8월20일, 수정완료일: 2010년4월2일

으로 평가되고 있다^[1-2]. H.264/AVC가 MPEG-4 Visual보다 50% 가까이 부호화 효율이 향상됨을 보임에 따라, 텔레비전을 위한 방송부터, 모바일 단말기와 같은 분야에 까지 폭넓게 사용되고 있다^[3-4]. 현재 우리나라의 지상파 DMB와 위성 DMB뿐만 아니라 유럽의 DVB-H, 미국 켈컴의 MediaFLO와 다양한 이동 멀티미디어 방송 기술에서 H.264/AVC를 비디오 압축을 위한 표준으로 채택하였다. 또한, H.264/AVC는 차세대 DVD와 IPTV를 위한 비디오 압축 표준으로도 사용될 예정이다.

이러한 H.264/AVC의 초기 버전에는 Baseline, Extended, Main의 3개의 프로파일이 제공되었다. 프로파일이란 개념은 MPEG-2부터 사용된 것으로 해당 코덱이 다양한 애플리케이션에서 활용될 수 있도록, 코덱이 지원하는 여러 틀들에 대해 복잡도와 성능을 고려하여 여러 개의 집합으로 구성시켜 놓은 것을 의미한다. H.264/AVC의 Baseline 프로파일은 가장 복잡도가 낮은 프로파일로 모바일 장치들에 주로 사용되고 있고, Main 프로파일은 복잡도는 높지만 고효율의 부호화 성능을 갖고 있어 방송 및 저장을 위한 애플리케이션에 사용되고 있다. 2005년 3월에는 H.264/AVC를 이용하여 HD급 영상에 대한 효과적인 부호화 위하여 FRExt (Fidelity Range Extension)라 불리는 확장 프로파일들이 추가되었고, 2007년 11월에는 스케일러빌리티 (Scalability) 기능을 지원하는 프로파일들이 추가되었다^[5-6].

H.264/AVC는 기존의 비디오 코덱보다 부호화 효율은 향상되었지만, 다중 참조 영상, 움직임 보상 과정에서의 높은 정밀도의 보간 필터, 다양한 블록 크기의 움직임 예측 모드, 인 루프 (in-loop) 디블록킹 필터, 적응적인 엔트로피 기술 등을 사용함에 따라 기존의 비디오 코덱들보다 복잡도가 높다. 높은 복잡도에 대한 요인은 코덱을 구현하는 환경과 최적화 단계에 따라 다르지만, 일반적으로 H.264/AVC의 부호화기 단에서는 움직임 예측 알고리즘이 주요한 요인이며, 복호화기 단에서는 인-루프 디블록킹 필터와, 움직임 보간 과정에서의 보간 필터링이 주요한 요인이 되고 있다^[7-13].

최근 MPEG에서는 HEVC (High Efficiency Video Coding)라는 이름으로 H.264/AVC 이후의 새로운 비디오 코덱을 개발하려는 움직임이 활발히 진행되고 있다. 이러한 차세대 비디오 코덱은 UDTV (Ultra Definition Television) 등에 사용될 수 있는 고화질·고품질의 코덱과 휴대전화와 같은 모바일 장치에서 효율적으로 사용

될 수 있도록 하는 고화질·저 복잡도의 코덱의 크게 두 가지 방향으로 연구가 진행되고 있다. 따라서 앞으로는 기존의 고화질·고품질을 위한 비디오 코덱 개발뿐만 아니라 코덱의 복잡도를 낮추면서 고효율의 부호화 성능을 갖는 비디오 코덱에 대한 연구가 매우 중요한 연구 분야가 될 것이다.

본 논문에서는 최근 급격히 사용량이 증가하고 있는 모바일 장치들을 위한 고화질·저 복잡도의 비디오 복호화기 설계에 대해 다룬다. 이를 위해 본 논문에서는 현재 가장 부호화 성능이 우수한 H.264/AVC의 복호화기 복잡도를 분석하였고, 복호화기 단에서 복잡도가 높은 디블록킹 필터에 대해 부호화 효율의 저하 없이 복잡도를 낮추는 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. II장에서는 H.264/AVC의 디블록킹 필터 알고리즘에 대해 기술하고, 기존의 저 복잡도 디블록킹 필터 알고리즘들을 소개한다. III장에서는 제안하는 CBP 값을 이용한 BS 값 계산 알고리즘과 강한 필터링의 필터링 범위 제한에 대해 설명한다. 또한, 이러한 기술들을 이용하여 설계한 하이브리드 저 복잡도 디블록킹 필터 기술에 대해 기술한다. IV장에서는 제안하는 알고리즘들에 대해 평가 및 분석하고, V장에서는 본 논문에 대한 결론 및 향후 연구의 진행방향을 제시하고 논문을 마치도록 한다.

II. 기존의 디블록킹 필터 방법

H.264/AVC의 복호화기는 움직임 보상 과정의 1/4-픽셀 보간 필터, 인 루프 디블록킹 필터, 그리고 적응적인 엔트로피 복호화 알고리즘 등의 사용 때문에, 기존의 MPEG-2, MPEG-4 비디오 복호화기들에 비해 상대적으로 높은 복잡도를 갖고 있다. 비디오 복호화기의 복잡도는 측정 조건 및 동작 환경에 따라 달라질 수 있지만, H.264/AVC의 복호화기의 복잡도는 대체적으로 인 루프 디블록킹 필터 (33%), 1/4-픽셀 보간 (25%), 비트 스트림 파싱 및 엔트로피 복호화 (13%), 역변환과 복원과정 (13%) 등으로 이루어진다^[14].

H.264/AVC의 디블록킹 필터는 복호화된 비디오의 주관적 화질 향상을 위하여 사용하던 비표준의 후처리 필터와 달리 비디오 복호화기 내부에 포함된 인 루프 형태이기 때문에, 부호화기에서도 복호화기와 동일한 디블록킹 필터를 사용해야만 한다^[15]. 이렇게 부호화기 단에서 디블록킹 필터를 사용함으로써 양자화 과정에서

표 1. H.264/AVC 디블록킹 필터의 부호화 성능
Table 1. Encoding efficiency of the H.264/AVC deblocking filter.

영 상		QP 22, 27, 32, 37	
		BDBitrate (%)	BDPSNR (dB)
CIF	Foreman, (300)	6.97	-0.29
	Hall_monitor, (300)	1.76	-0.09
	Tempete, (260)	1.81	-0.08
	Paris, (300)	4.46	-0.24
평균		3.75	-0.18

발생하는 블록킹 에러를 제거할 수 있고, 이를 통해 주관적 화질을 향상시킬 수 있다. 또한, 블록킹 에러가 제거된 프레임을 참조 프레임으로 사용하여 움직임 예측 및 보상 과정을 수행함으로써 더 정확한 예측이 가능하고, 이에 따라 부호화 효율도 향상될 수 있다.

표 1은 H.264/AVC의 참조 소프트웨어인 JM 12.4에 디블록킹 필터를 적용한 것과 적용하지 않은 것에 대한 BDPSNR과 BDBitrate 결과 값이다^[16-17]. 이 실험에는 CIF (Common Intermediate Format) 크기의 영상 6개를 사용하였으며, 실험 조건은 VCEG에서 요구하는 공통 조건 (Common condition)의 Baseline 프로파일 설정을 이용했다^[18]. 실험 결과, 디블록킹 필터를 적용하지 않을 때에 같은 부호화 조건에서 디블록킹 필터를 적용한 것에 비해 평균 비트율이 약 3.75% 증가하였음을 확인할 수 있었다. 추가로 인터 모드의 탐색 방법과 인트라 모드의 예측 모드 제한과 같은 부호화 설정 값을 변경하여 실험한 경우에는 CIF 영상 6개에 대해 최대 14%, 평균 6.8% 가까이 비트율이 증가함을 확인하였다.

H.264/AVC의 디블록킹 필터는 4x4 블록 또는 8x8 블록 단위의 이산여현변환 과정 이후 수행되는 양자화 과정에서 각 블록의 경계에 발생하는 블록킹 에러를 제거하기 위해 사용된다. 그림 1은 4x4 블록 단위로 이산여현변환을 사용한 경우의 휘도 성분에 대한 매크로블록 내부에서의 필터링 될 경계 위치를 보여준다. H.264/AVC의 디블록킹 필터링은 수직 에지들에 대해 왼쪽에서 오른쪽으로 순서대로 필터링을 먼저 수행한 후, 수평 에지들에 대해 위쪽에서 아래쪽으로 순서대로 필터링을 수행한다. 매크로블록에서 8x8 블록 단위의 이산여현변환을 사용한 경우에는 그림 1의 실선 부분과

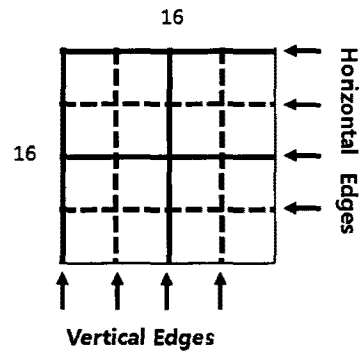


그림 1. 매크로블록 내부의 필터링 될 경계들
Fig. 1. Boundaries in a macroblock to be filtered.

같이 8x8 블록의 경계 지점에서만 디블록킹 필터링을 수행한다. 이러한 H.264/AVC의 디블록킹 필터 과정은 크게 블록 경계를 기준으로 BS (Boundary Strength) 값을 계산하는 부분과 계산된 BS 값을 이용하여 필터링을 적용하는 두 부분으로 나뉜다.

1. BS 값 계산부

H.264/AVC의 디블록킹 필터는 블록 경계에 대해 BS 값을 계산하고, 그 BS 값에 따라 필터를 선택적으로 적용한다. 블록 경계에서의 BS 값은 블록 경계를 포함하는 매크로블록들의 예측 모드, 두 블록 내의 0이 아닌 변환 계수 값의 존재 여부, 두 블록의 움직임 벡터

표 2. BS 값의 결정 조건들
Table 2. Decision conditions of BS parameter.

블록 모드 및 조건들	BS
P 또는 Q 블록이 인트라 예측 모드이고, 에지가 매크로블록 경계에 있는 경우	4
P 또는 Q 블록이 인트라 예측 모드이고, 에지가 매크로블록 경계에 있지 않는 경우	3
P 와 Q 블록이 모두 인터 예측 모드이고, P 또는 Q 블록에 부호화된 계수 값이 존재하는 경우	2
P 와 Q 블록이 모두 인터 예측 모드이고, P 와 Q 블록에 부호화된 계수 값이 존재하지 않고, 움직임 벡터 차분 값이 1보다 크거나 참조 프레임 인덱스가 서로 다른 경우	1
P 와 Q 블록이 모두 인터 예측 모드이고, P 와 Q 블록에 부호화된 계수 값이 존재하지 않고, 움직임 벡터 차분 값이 1보다 작고, 프레임 인덱스가 서로 같은 경우	0

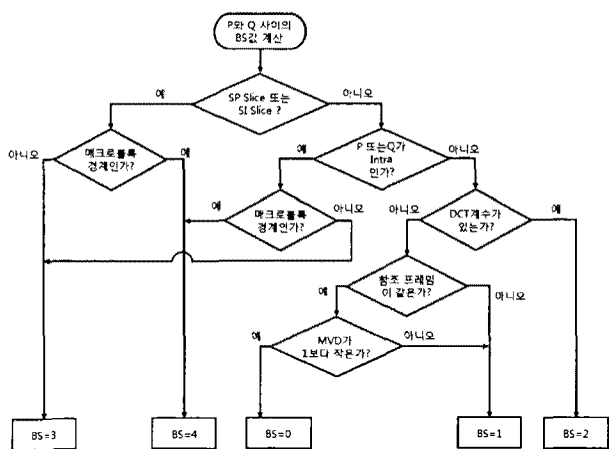


그림 2. BS 값의 결정 구조
Fig. 2. Decision structure of BS value.

의 차이 값, 두 블록의 참조 프레임 인덱스 값에 따라 계산된다. 블록 경계에서의 BS 값은 블록 경계를 기준으로 왼쪽 또는 위쪽에 있는 블록을 P, 오른쪽 또는 아래쪽에 있는 블록을 Q라 할 때, 앞에서 설명한 4개의 파라미터 값에 따라 표 2와 같이 결정된다.

BS 값의 결정 과정은 4개의 파라미터 값에 따라 결정되기 때문에, 이러한 값들을 비교하는 과정에서 많은 분기문을 필요로 한다. 그림 2는 JM 12.4에 구현된 BS 값의 결정 과정을 나타낸 것으로, 그림에서 보이는 것과 같이 각각의 BS 값을 결정하기 위해 많은 분기문이 필요하다는 것을 확인할 수 있다.

예를 들어, 인트라 슬라이스 내에 존재하는 P와 Q블록에 대한 BS 값의 결정 과정에는 평균적으로 3개의 분기문이 필요하다. 그리고 P 슬라이스에서 BS 값이 2인 경우는 3개, BS 값이 1인 경우에는 4.5개, BS 값이 0인 경우에는 5개의 분기문이 필요하다.

2. 필터링 적용부

필터링 적용부에서는 P와 Q블록의 픽셀 값에 대해 계산된 BS 값에 따라 필터링을 적용적으로 적용한다. 그림 3과 같이 BS 값이 0인 경우에는 필터링을 적용하지 않고, BS 값이 4인 경우에는 강한 필터링을 적용하며, 나머지 BS 값에 대해서는 보통 필터링 (Normal filtering)을 적용한다. 이렇게 필터링 적용부에서는 BS 값에 따라 서로 다른 특성의 필터를 사용하며, 필터를 적용할 때에도 해당 픽셀 값에 따라 필터링 적용 여부를 다시 판단하는 이중 결정 구조로 되어 있다. 따라서 블록 경계에서 BS 값이 1~4 중 하나의 값이 선택되더라도 픽셀의 임계값에 따라 필터링이 수행되지 않을 수

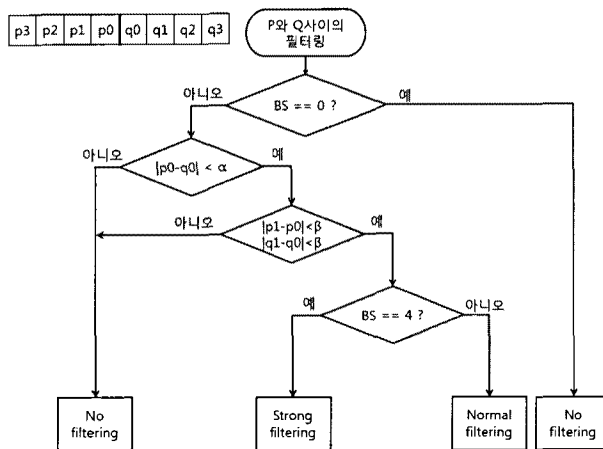


그림 3. 필터링 적용부의 결정 구조
Fig. 3. Decision structure of filtering application.

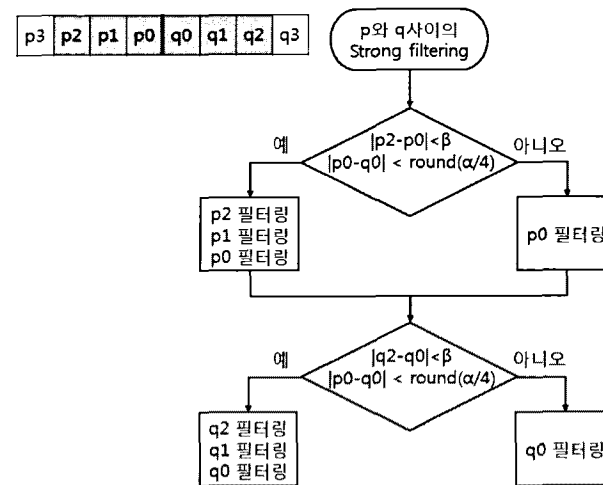


그림 4. Strong filtering의 구조
Fig. 4. Structure of strong filtering.

있다.

BS 값이 4인 경우는 P와 Q블록이 인트라 매크로블록 경계에 있는 것으로, 다른 BS 값보다 상대적으로 강한 필터링을 적용한다. 이 경우 그림 4와 같이 픽셀 값의 차이 값을 미리 계산된 파라미터인 α , β 와 비교한 후 필터링의 적용 범위를 결정한다. 조건을 만족하는 경우에는 에지 경계를 기준으로 좌·우 세 픽셀까지 필터링을 적용하며, 이 경우 각 픽셀에 적용되는 필터의 특성은 서로 다르다. 조건을 만족하지 않을 때에는 에지 경계를 기준으로 좌·우 한 픽셀에 대해서만 필터링을 적용한다. 이러한 과정은 수평 방향의 에지에 대한 필터링 과정에도 동일하게 적용된다.

두 블록 경계에서의 BS 값이 1, 2, 3 중 하나일 때는 해당 경계에 보통 필터링이 적용된다. 이 경우에도 필터링을 적용할 위치의 픽셀 값의 차이 값을 β 와 비교하

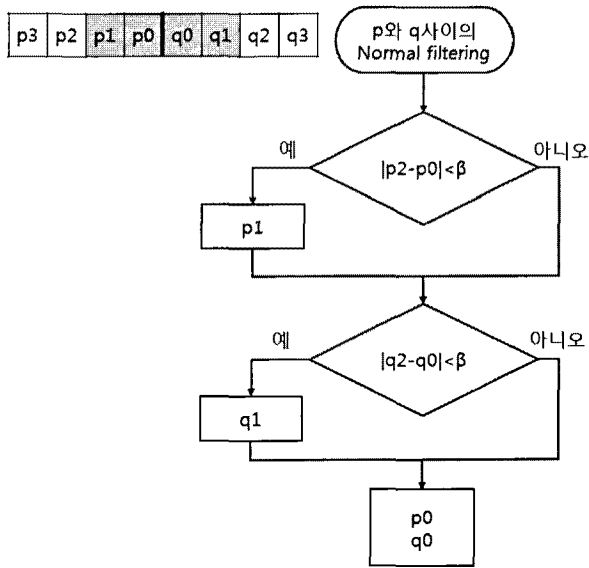


그림 5. Normal filtering의 구조
Fig. 5. Structure of normal filtering.

고 나서 필터링의 적용 범위를 결정한다. 그림 5의 구조에 따라 조건을 만족하지 않을 때에는 p0만 필터링을 적용하고, 조건을 만족할 때에는 p0, p1의 두 픽셀에 필터링을 적용한다. 이러한 과정은 Q블록의 픽셀 q0, q1에도 적용되며, 그림 5의 보통 필터링 과정은 수평 방향의 에지에 대한 필터링 과정에도 동일하게 적용된다.

3. 기존의 저 복잡도 디블록킹 필터

H.264/AVC의 디블록킹 필터의 복잡도를 낮추는 연구 중 J. Ren은 디블록킹 필터 연산 과정의 BS 값 계산부와 필터링 적용부에 대해 복잡도를 낮추는 방법을 제안하였다^[19]. H.264/AVC의 디블록킹 필터는 BS 값이 0인 블록 경계에는 디블록킹 필터링을 수행하지 않는다. 해당 논문에서는 SKIP 모드 타입의 매크로블록은 매크로블록 내에 존재하는 블록들의 BS 값이 모두 0으로 결정되는 것을 보였다. 이러한 특성을 이용하여 매크로블록이 SKIP 모드 타입이면 해당 매크로블록에 디블록킹 필터링을 수행하지 않음으로써 복호화기의 연산량을 감소시켰다. 또한, 추가적인 연산량 감소를 위해 매크로블록 타입이 SKIP 모드가 아닌 경우에는 BS 값 계산 과정의 BS 값의 계산 횟수를 감소시켰다. 이 방법은 매크로블록 내의 4x4 블록 경계에 대해 BS 값을 모두 계산하는 것이 아니라 일부 경계에서만 BS 값을 계산하고, BS 값을 계산하지 않는 경계에서는 인접하는 경계에서 계산한 BS 값을 그대로 이용하도록 하였다. 그러나 이 방법은 BS 값을 계산하지 않고 인접하는 블록

경계의 BS 값을 사용하기 때문에, BS 값의 예측 에러가 크게 발생할 수 있는 문제가 있다.

디블록킹 필터의 연산량 감소를 위해 Jian Lou는 H.264/AVC의 High 프로파일에 대해 통계적 특성을 이용하여 BS 값 결정 과정에 효과적인 허프만 트리 구조를 제안하였다^[20]. BS 값을 계산하는 과정은 많은 분기문을 가지는데, 이러한 분기문들은 분기 예측이 실패하는 경우 파이프라인 내의 모든 명령어를 버리게 하므로 파이프라인 구조를 갖는 프로세서의 동작 성능을 떨어트리게 된다. 이 논문에서는 High 프로파일의 각 슬라이스 구조에 대한 실험을 통해 BS 값의 발생 빈도를 확인하였다. 이를 통해 자주 발생하는 BS 값의 결정 조건을 조건문에서 먼저 검사함으로써, BS 값을 계산하는데 소요되는 평균 분기문의 수 및 복호화기의 연산량을 감소시켰다.

III. 제안하는 복호화 방법

본 논문에서는 비디오 복호화기의 복잡도 감소를 위한 저 복잡도 디블록킹 필터 알고리즘을 제안한다. 기존의 H.264/AVC의 디블록킹 필터는 이산여현변환 블록의 경계 영역에서 BS 값을 구한 후, 계산된 BS 값에 따라 필터링을 수행하였다. 그러나 이러한 BS 값은 경계 지점에 있는 블록들의 문맥 정보들에 따라 결정되는데, 주변 문맥 정보들을 참조하는 과정에서 많은 분기문을 필요로 한다. 본 논문에서는 디블록킹 필터의 BS 값의 계산 부분을 8x8 블록 단위의 CBP (Coded Block Pattern) 값을 이용하여 계산하도록 대체하여, 기존의 트리 구조 기반의 BS 값 결정 알고리즘 보다 분기문의 수를 감소시키고, 이를 통해 복잡도를 낮추는 방법을 3.1절에서 제안한다. 또한, 인트라 매크로블록 경계에서 적용되는 강한 필터링의 필터링 범위를 기존의 세 픽셀에서 두 픽셀로 제한함으로써 추가적인 복잡도를 낮추는 방법을 3.2절에서 제안한다. 이렇게 제안된 두 가지의 기술에 추가로 J. Ren에 의해 제안된 SKIP 매크로블록에서는 디블록킹 필터링을 수행하지 않는 알고리즘을 결합하여 디블록킹 필터링 연산의 복잡도를 낮추는 하이브리드 저 복잡도 디블록킹 필터 알고리즘에 대해서 3.3절에서 소개한다.

본 논문에서는 저 복잡도 복호화기 설계에 대해서만 다루고 있지만, 저 전력으로 동작하는 저 복잡도 부호화기 설계를 위해 H.264/AVC의 참조 소프트웨어인 JM

표 3. H.264/AVC 인코딩 환경 설정

Table 3. H.264/AVC encoding configure setting.

파라미터 이름	파라미터 값
ProfileIDC	66 (Baseline 프로파일)
LevelIDC	4.0
Search Range	±16 픽셀
NumberReferenceFrames	3 장
Inter mode	InterSearch8x4 (off) InterSearch4x8 (off) InterSearch4x4 (off)
Intra mode	Intra4x4DirDisable (on)
Coding structure	IPPP 구조
SymbolMode	0 (CAVLC 사용)
RDOptimization	2 (RDO on Fast High complexity mode)
Transform8x8Mode	0 (only 4x4 transform)
SearchMode	3 (EPZS patterns)

12.4에 대해 표 3과 같은 제한된 조건에서 동작시킴으로써 부호화기의 복잡도를 감소시켰다. 표 3의 조건은 매크로블록의 화면 간 예측 모드에서 8x8블록 이하의 서브 블록 예측 모드를 사용하지 않음으로써 부호화기에서의 움직임 예측 과정의 복잡도를 낮춘다. 또한, 4x4 블록 사이즈의 화면 내 예측에서도 대각선 모드를 제한하여 확률적으로 자주 나오는 인트라 예측 모드만을 사용하여 화면 내 예측 과정의 복잡도를 낮춘다. 이러한 조건을 LMBT (Limited Macro-Block Type)라 하며, 저 복잡도 부호화기에서는 이 조건에 따라 영상을 부호화함으로써 부호화 효율을 최대한 유지하면서, 부호화기의 복잡도를 낮출 수 있다.

1. CBP 값을 이용한 BS 값 계산

LMBT 조건을 사용하는 비디오 부호화기는 부호화기의 복잡도 감소를 위하여 화면 간 예측 시 8x8 블록 크기 이하의 서브 블록모드의 움직임 예측을 수행하지 않도록 설계하였다. 그러므로 8x8 블록 내에 있는 4개의 4x4 블록들은 참조 프레임 인덱스와 움직임 벡터 값

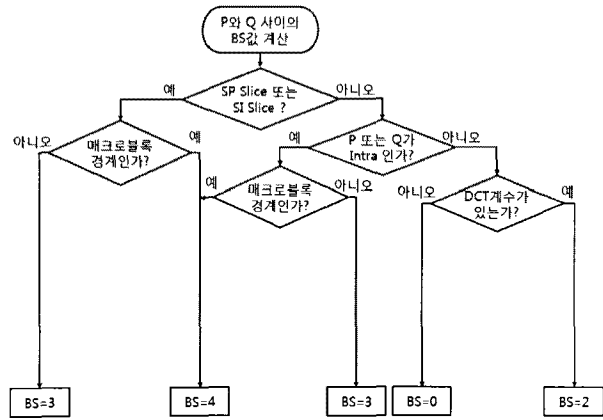


그림 6. 간소화된 BS 값의 결정 구조
Fig. 6. Simplified decision structure of BS value.

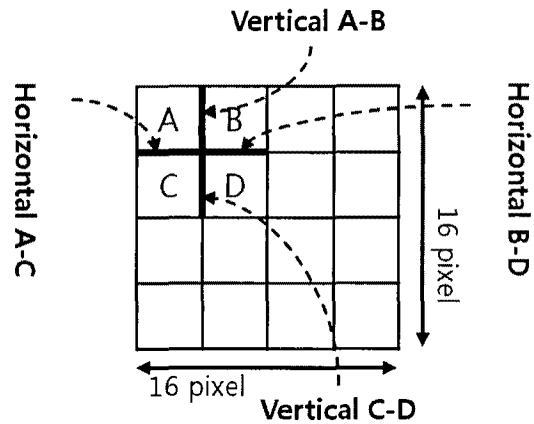


그림 7. 8x8 블록내의 BS 값 계산 위치들
Fig. 7. Calculated positions of BS in 8x8 block.

이 같게 된다. 따라서 8x8 블록 내에 있는 4개의 4x4 블록 경계에서의 BS 값의 계산 과정은 그림 6과 같이 두 블록의 변환 계수 값의 존재 여부만을 통해 계산되도록 간소화될 수 있다. 그러나 여전히 8x8 블록들 간의 경계나 매크로블록들 간의 경계에 대해서는 그림 2와 같은 기존의 BS 값의 결정 구조를 사용해야 한다.

그림 7은 매크로블록에서 첫 번째 8x8 블록 내의 4개의 4x4 블록 (A, B, C, D)들과 그 경계에 있는 4개의 BS 값 계산 위치에 대한 것이다. Vertical A-B, Vertical C-D, Horizontal A-C, Horizontal B-D에 대한 BS 값은 앞에서 설명한 것처럼 그림 6의 간소화된 BS 값의 결정 구조를 통하여 계산될 수 있다. 마찬가지로 매크로블록 안의 나머지 8x8 블록들의 내부에 BS 값들도 동일하게 계산될 수 있다. 그러나 여전히 이와 같은 간소화된 BS 값의 결정 구조에도 다수의 분기문이 존재한다.

III장 1절에서 제안하는 CBP 값을 이용한 BS 값 계

표 4. CBP 값을 이용한 BS 값 계산 알고리즘의 의사코드

Table 4. Pseudocode of BS value calculation algorithm using CBP value.

```

iff( inter macroblock )
{
    bs_0 = ( cbp & 1 ) << 1
    bs_1 = ( cbp & (1<<1) )
    bs_2 = ( cbp & (1<<2) )>>1
    bs_3 = ( cbp & (1<<3) )>>2
}
else
{
    bs_0 = bs_1 = bs_2 = bs_3 = 3
}
    
```

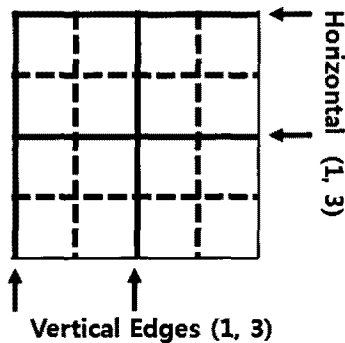


그림 8. 매크로블록 경계, 8x8 블록 경계, 8x8 블록 내부 경계
 Fig. 8. Macroblock boundary, 8x8 block boundary and 8x8 block inner boundary.

산 알고리즘은 8x8 블록 내에서 계산해야 할 4개의 BS 값을 그림 6의 간소화된 BS 결정 구조를 이용하여 각각 구하는 것이 아니라, 8x8 블록 단위에서 CBP 값을 이용하여 동시에 4개의 BS 값을 계산한다. 표 4는 제안하는 알고리즘의 의사코드(Pseudocode)이다. 제안하는 알고리즘은 매크로블록에서 4개의 8x8 블록들 내부에 존재하는 16개의 BS 값 계산 위치에 대해, 하나의 분기문으로 BS 값을 동시에 계산할 수 있다.

제안된 CBP 값을 이용한 BS 값 계산 알고리즘을 적용한 디블록킹 필터는 그림 8과 같이 매크로블록의 수평, 수직 방향에 대해 첫 번째와 세 번째에 있는 에지에서의 BS 값의 계산은 그림 2에서 살펴본 기존의 BS 값 결정구조를 그대로 이용한다. 그러나 그림 8에서 점선

표 5. BS 값 계산과정의 분기문 수 비교

Table 5. Comparison of number of branch instruction for BS calculation.

구 분	매크로블록 단위의 분기문의 수
기존 알고리즘 (JM12.4)	$3.8 \times 32 = 121.6$
제안된 알고리즘	$3.8 \times 16 + 2 = 62.8$

으로 표시된 두 번째와 네 번째 경계에 있는 에지에서의 BS 값의 계산은 BS 값 결정 구조를 이용하지 않고 표 4의 의사코드를 이용한다.

표 5는 JM 12.4에 구현된 기존의 BS 값 계산 과정에서 사용하는 분기문의 수와 제안하는 알고리즘이 사용하는 분기문의 수를 매크로블록 단위에서 비교한 것이다.

P 슬라이스에서는 인트라 매크로블록과 인터 매크로블록이 모두 사용될 수 있으므로, BS 값은 0~4까지의 값이 모두 발생한다. BS 값이 4인 경우에는 그림 2의 트리 구조에서 3개의 분기문이 사용된다. BS 값이 3인 경우에도 3개의 분기문이 사용된다. BS 값이 0인 경우에는 5개의 분기문이 사용되며, BS 값이 1인 경우에는 조건에 따라 5개와 4개의 분기문이 사용된다. BS 값이 2인 경우에는 3개의 분기문이 사용된다. 그러므로 매크로블록 내의 하나의 4x4 블록 경계에 대한 BS 값 계산에는 평균 3.8개의 분기문이 사용된다. 따라서 매크로블록 내에서 32개의 BS 값을 모두 계산하기 위해서는 기존의 알고리즘에서는 121.6개의 분기문이 필요하다.

그러나 제안하는 CBP 값을 이용한 BS 값 계산 알고리즘을 이용하는 경우에는 8x8 블록 경계에서는 기존의 알고리즘을 사용하지만, 8x8 블록 내부에 대해서는 CBP 값을 통해 16개의 BS 값을 계산하므로 한 개의 분기문만이 필요해진다. 그러나 그림 2와 같이 기존의 BS 결정 구조에서 SP 슬라이스 또는 SI 슬라이스를 검사하는 분기문이 사용됨을 고려하여 매크로블록 당 2개의 분기문이 필요한 것으로 간주하였다. 따라서 매크로블록 내의 8x8 블록 경계에서 BS 값을 계산하는데 60.8개의 분기문이 필요하고, 블록 내부의 BS 값을 계산하는데 2개의 분기문이 필요하기 때문에 전체적으로 62.8개의 분기문이 필요하다.

CBP 값을 이용하여 BS 값을 계산하는 알고리즘은 그림 6과 같이 블록 내의 변환 계수 값의 존재 여부를 확인하여 BS 값을 계산하는 것에 비해 정확한 BS 값을

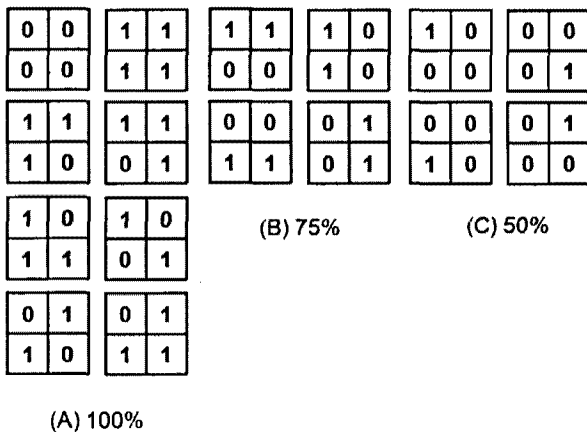


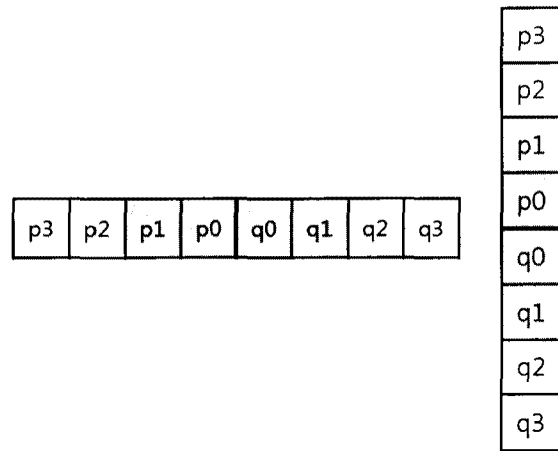
그림 9. 8x8 블록내의 변환 계수 값의 패턴
Fig. 9. Patterns of transform coefficient in 8x8 block.

계산하지는 못한다. CBP 값을 이용하여 BS 값을 계산하는 경우, 8x8 블록 내의 4개의 4x4 블록들의 변환 계수 값의 패턴과 BS 값의 예측률은 그림 9와 같다.

그림 9의 (A)패턴의 경우는 BS 값을 계산할 때 4x4 블록들의 변환 계수 값을 확인하는 방법과, CBP 값만으로 4개의 BS 값을 동시에 구하는 방법의 BS 값이 같아진다. 그림 9의 (B)패턴인 경우에는 8x8 블록내의 4개의 BS 값 중 1개의 BS 값에 대해 예측이 실패하며 (C)패턴인 경우에는 8x8 블록 내에서 2개의 BS 값의 예측이 실패한다. 이를 통해 8x8 블록의 CBP 값으로 BS 값을 계산하는 경우의 에러율은 평균 18.7%임을 확인할 수 있다. H.264/AVC의 디블록킹 필터는 앞에서 살펴보았듯이 계산된 BS 값에 따라 고정적으로 필터링을 적용하는 것이 아니라, 필터링을 적용할 경계 주변의 픽셀 값에 따라 필터링의 적용 여부를 재결정한다. 즉, CBP 값을 이용하여 BS 값을 계산하는 알고리즘을 사용하는 경우 BS 값이 0인 경계에 대해 BS 값을 2로 예측하는 경우가 발생할 수 있지만, 필터링을 적용할 때 픽셀 값을 이용하여 필터링의 적용 여부를 재판단하기 때문에 실제 발생하는 에러율은 18.7% 보다 더 낮아질 수 있다. 이러한 제안된 방법은 BS 값의 예측 에러율이 단순히 주변의 BS 값을 복사하여 사용하는 J. Ren의 방법보다 낮아서 예측 오류에 따른 화질 저하가 크지 않다는 장점이 있다.

2. 인트라 매크로블록의 필터링 적용 범위 제한

H.264/AVC의 디블록킹 필터는 인트라 매크로블록 경계에 대해 강한 필터링을 적용한다. 강한 필터링을 적용하는 경우의 필터링의 적용 범위는 그림 4와 같이



(a) 수평 방향 (b) 수직 방향

그림 10. BS=4인 경우의 필터링 적용 범위
Fig. 10. Range of filtering case of BS=4.

임계값 조건을 만족하는 경우, 경계를 기준으로 최대 세 픽셀까지이다. 본 연구에서는 복호화기의 복잡도 감소를 위해 강한 필터링을 적용하는 때도 필터링의 적용 범위를 그림 10과 같이 수평, 수직 방향에 대해 두 픽셀로 제한하였다. 이를 통해 부호화 효율을 유지하면서, 복호화기의 복잡도를 감소시켰다.

3. 하이브리드 저 복잡도 디블록킹 필터

본 절에서 소개하는 하이브리드 저 복잡도 디블록킹 필터는 J. Ren이 제안한 SKIP 모드 매크로블록에 대해 디블록킹 필터를 수행하지 않는 알고리즘과 본 연구에서 제안한 CBP 값을 이용한 BS 값 계산 단순화 알고리즘, 인트라 매크로블록의 필터링 범위 제한 알고리즘을 결합하여 사용한다. J. Ren이 제안한 알고리즘은 상대적으로 높은 QP 범위에서 SKIP 매크로블록 모드의 발생 빈도가 높은 경우에 대하여 연산 복잡도 감소의 효과가 크다. 즉, J. Ren의 방법은 영상의 통계적인 특성을 바탕으로 하는 알고리즘이며, 이에 비해 본 논문에서 제안한 CBP 값을 이용한 BS 값 계산 단순화 알고리즘과 인트라 매크로블록의 필터링 적용 범위 제한 알고리즘은 영상의 통계적 특성과 관계없이 복잡도를 감소시키는 방법이다. 이렇게 서로 다른 특성을 갖는 알고리즘들을 결합한 하이브리드 저 복잡도 디블록킹 필터는 그림 11과 같이 먼저 필터링을 수행할 때 매크로블록 모드를 검사하여 SKIP 모드인 경우에는 해당 매크로블록에는 디블록킹 필터링을 수행하지 않는다. 그리고 현재 매크로블록이 SKIP 모드가 아닌 경우에는

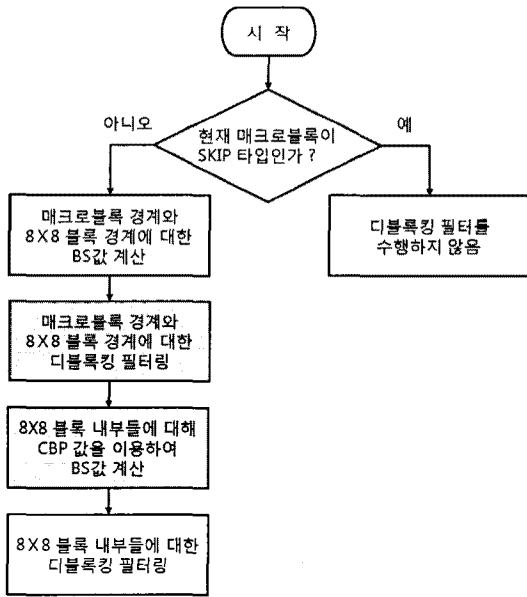


그림 11. 저 복잡도 디블록킹 필터의 순서도
Fig. 11. Flowchart of low-complexity deblocking filter.

매크로블록의 경계와 8x8 블록 경계에 대해 그림 2의 기존 BS 값 결정 트리 구조를 이용하여 BS 값을 계산한다. 이렇게 계산된 BS 값에 따라 해당 경계에 대해 수직, 수평 방향으로 필터링을 적용한다. 이러한 필터링 적용 과정에서 강한 필터링을 이용하는 경우(BS=4)에는 그림 10과 같이 필터링의 적용 범위를 두 픽셀로 제한한다.

매크로블록에서 아직 디블록킹 필터링이 수행되지 않는 8x8 블록 내부들에 대해서는 표 4에서 소개한 의사코드를 적용하여 CBP 값만으로 16개의 BS 값을 한번에 계산한다. 이러한 방법을 통해 기존에 평균 60.8개의 분기문이 필요했던 부분에 대해 두 개의 분기문만으로 BS 값을 계산할 수 있게 된다. 이렇게 계산된 8x8 블록 내부의 블록 경계에 대한 BS 값을 이용하여 해당 경계에서 수직, 수평 방향으로 필터링을 적용함으로써 매크로블록 전체에 대한 디블록킹 필터링 과정은 종료된다. 이러한 제안된 하이브리드 디블록킹 필터는 LMBT 조건을 사용하는 부호화기에서도 동일하게 적용되기 때문에, 부호화기와 복호화기 사이의 오차는 발생하지 않는다.

IV. 실험 및 결과

1. 실험 환경

제안하는 저 복잡도 디블록킹 필터의 성능을 평가하

기 위하여 H.264/AVC의 참조 소프트웨어 JM 12.4에 그림 11의 알고리즘을 적용하였다. 이를 이용하여 제안하는 디블록킹 필터를 사용하는 경우의 부호화기의 부호화 성능과 복호화기의 복호화 성능을 평가하였다. 먼저 부호화기의 부호화 성능은 BDBitrate와 BDPSNR값을 이용하여 평가하였으며, 복호화기의 복호화 성능은 수식 (1)의 ATS (Average Time Saving)를 이용하여 평가하였다.

$$ATS(\%) = \frac{refDectime - testDectime}{refDectime} \times 100 \quad (1)$$

수식 (1)에서 *refDectime*은 참조 소프트웨어인 JM 12.4에 LMBT 조건을 적용하여 영상을 부호화한 후 복호화하여 측정된 복호화 시간이다. *testDectime*은 참조 소프트웨어에 제안하는 알고리즘을 구현하고서 LMBT 조건을 적용하여 영상을 부호화하고, 복호화하여 측정된 복호화 시간이다.

이러한 실험은 Intel Core 2 Duo 1.6GHz CPU, 2GB RAM의 PC 환경에서 수행되었으며, 모바일 환경에서 많이 사용하는 CIF 크기의 Foreman, Hall_monitor, Mobile, Paris, Coastguard, Container의 6개 영상을 이용하였다. 이러한 실험 영상 각각에 대해 4개의 QP 값 (22, 27, 32, 37)으로 300 프레임씩 부호화한 후, 복호화기를 통해 복호화함으로써 복호화 시간을 측정하였다. 이러한 복호화 과정은 부호화 과정보다 연산량이 적어 복호화 시간이 짧고, 이에 따라 측정 시마다 측정되는 복호화 시간의 분산 값이 크게 된다. 따라서 본 실험에서는 각각의 실험 영상에 대해 각 QP 값별로 100번씩 복호화를 수행하고 나서, 평균 복호화 시간을 계산하고 이 값을 통해 ATS 값을 계산하였다.

참조소프트웨어 JM 12.4에는 복호화 과정이 끝나고 나서 DPB(Decoded Picture Buffer)에 저장된 복호화된 프레임을 파일로 출력하며, 각 프레임에 대하여 PSNR 값을 계산하는 과정이 포함되어 있다. 그러나 이러한 과정은 실제 애플리케이션에서는 필요하지 않은 기능이며, 또한 파일을 쓰는 부분에 의해 복호화 시간 측정 과정에서 측정 시간의 오차가 크게 발생하는 문제가 발생한다. 따라서 본 논문에서는 정확한 복호화 시간 측정을 위하여 해당 부분들의 코드를 제거하고 나서 복호화 시간을 측정하였다.

2. 실험 결과

표 6은 그림 6의 간소화된 BS 값 결정 구조, J. Ren의 알고리즘, 제안하는 저 복잡도 디블록킹 필터에 대한 각각의 ATS 값을 보인다. 각 실험의 ATS 값 계산에 사용된 *refDecTime*은 JM12.4에 LMBT 조건을 적용하여 실험 영상을 부호화하고 나서, 복호화함으로써 측정된 것이다. 제안하는 알고리즘을 사용하는 경우 참조 복호화기 보다 CIF 크기 영상 6개에서 복호화 성능이 최대 21.02%, 평균 8.96% 향상되었으며 디블록킹 필터에서의 연산량 감소는 평균 42% 향상되었다. 제안하

표 6. Simplified 방법, J. Ren 방법, 제안된 방법의 ATS 값

Table 6. ATS values of simplified method, J. Ren method and proposed method.

영 상	QP	Simplified method	J. Ren method	Proposed method
		ATS (%)	ATS(%)	ATS (%)
Foreman	22	-0.77	0.46	1.91
	27	0.29	3.01	4.29
	32	0.62	5.38	7.24
	37	1.71	9.56	10.48
Hall_monitor	22	3.59	6.37	5.48
	27	6.06	14.61	13.44
	32	6.95	20.11	18.87
	37	7.58	22.40	21.02
Mobile	22	-1.53	-0.29	0.18
	27	-1.17	0.12	0.90
	32	-0.24	1.43	2.79
	37	0.52	5.01	6.09
Paris	22	0.01	5.91	6.49
	27	0.94	8.05	8.53
	32	1.65	10.37	11.12
	37	3.22	13.70	14.60
Coast guard	22	1.68	1.91	2.56
	27	2.74	3.26	4.22
	32	4.28	5.82	7.14
	37	5.87	10.47	12.10
Container	22	2.10	6.80	7.60
	27	3.65	11.59	12.57
	32	4.87	16.00	16.66
	37	5.44	19.61	18.82
AVERAGE		2.50	8.40	8.96

표 7. J. Ren 방법, 제안된 방법의 BDBitrate와 BDPSNR

Table 7. BDBitrate and BDPSNR of J. Ren method and proposed method.

영 상	J. Ren method		Proposed method	
	BDBitrate (%)	BDPSNR (dB)	BDBitrate (%)	BDPSNR (dB)
Foreman	0.36	-0.02	0.00	0.00
Hall_monitor	0.67	-0.01	0.23	0.00
Mobile	-0.01	0.01	-0.14	0.01
Paris	0.28	-0.02	-0.07	0.00
Coastguard	0.50	-0.02	0.07	0.00
Container	0.05	0.00	-0.22	0.01
Average	0.29	-0.01	-0.02	0.00

는 알고리즘은 일부 경계에서만 BS 값을 계산하는 J. Ren의 알고리즘과 비슷한 성능을 보인다.

표 7은 제안하는 저 복잡도 디블록킹 필터를 사용하는 경우의 부호화 성능을 실험한 결과로 JM 12.4에 LMBT 조건을 적용하는 참조 부호화기와 비교를 한 것이다. 디블록킹 필터의 연산량을 감소시키기 위하여 디블록킹 필터를 간소화하는 경우 부호화기의 성능이 저하될 수 있다. 그러나 제안하는 알고리즘을 사용하는 경우 CIF 영상 6개에 대해 평균 BDBitrate가 0.02% 감소한 것을 확인할 수 있다. 이는 제안하는 알고리즘의 부호화 성능이 참조 부호화기 보다 0.02% 향상된 것을 의미한다. 또한, 제안하는 알고리즘은 J. Ren의 알고리즘에 대해서는 부호화 성능이 0.3% 향상되었다.

V. 결론 및 향후 연구 계획

본 논문에서는 저 복잡도 비디오 복호화기를 위해 디블록킹 필터의 연산량을 낮추는 알고리즘을 제안하였다. H.264/AVC의 디블록킹 필터링 연산은 BS 값을 계산하는 과정에서 많은 분기문을 사용하고, 이에 따라 복호화기의 연산량이 높다. 본 논문에서는 매크로블록의 8x8 블록 경계에서는 기존의 H.264/AVC의 BS 값 계산 알고리즘을 그대로 사용하고, 8x8 블록 내부의 BS 값을 계산하는 것에 대해 CBP 값만을 이용하여 계산하는 알고리즘을 제안하였다.

이 방법은 기존의 BS 값 계산 알고리즘이 다수의 분기문을 사용하는 것에 비해, 한 개의 분기문 만으로 매

크로블록 내의 BS 값을 계산하여 매크로블록 당 BS 값 계산에 사용되는 분기문의 수를 49% 감소시켰다. 또한, BS 값이 4인 경우 적용하는 강한 필터링에 대해서도 필터링의 적용 범위를 최대 두 픽셀로 제한함으로써 추가로 복호화기의 복잡도를 감소시켰다. 이러한 두 가지의 제안하는 기술과 J. Ren이 제안한 SKIP 모드의 매크로블록에 대해 디블록킹 필터를 수행하지 않는 알고리즘을 결합한 하이브리드 저 복잡도 디블록킹 필터는 참조 복호화기 보다 복호화기의 평균 복잡도는 8.96%, 디블록킹 필터 알고리즘에서는 42% 감소함을 보였고, 이에 비해 부호화 성능은 거의 변화가 없음을 확인하였다. 향후에는 H.264/AVC에서 복호화기에서 연산 복잡도가 높은 부분인 엔트로피 복호화 알고리즘과 1/4-픽셀 보간 필터 알고리즘에 대해 부호화 성능을 유지하면서 복호화기의 복잡도를 낮추는 연구를 진행할 것이다.

참고 문헌

- [1] A. Luthra, G. J. Sullivan, and T. Wiegand, "Introduction to the special issue on the H.264/AVC video coding standard," *IEEE Trans. Circuit Syst. Video technol.*, vol. 13, pp. 557-559, July 2003.
- [2] Draft IUT-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), Mar. 2003.
- [3] A. Joch, F. Kossentini, H. Schwarz, T. Wiegand, and G. Sullivan, "Performance comparison of video coding standards using lagrangian coder control," *Int.Conf.Image Processing(ICIP)*, vol. 2, pp. 501-504, Oct. 2002.
- [4] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. Sullivan, "Rate- constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 688-703, July 2003.
- [5] G. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions," *SPIE Conference on Applications of Digital Image Processing*, August, 2004.
- [6] H. Schwarz, D. Marpe, T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Trans. Circuit Syst. Video technol.*, vol. 17, pp. 1103-1120, Sept. 2007.
- [7] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video technol.*, vol. 13, pp. 560-576, July 2003.
- [8] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Trans. Circuits Syst. Video technol.*, vol. 13, pp. 657-673, July 2003.
- [9] S. Wenger, "H.264/AVC over IP," *IEEE Trans. Circuits Syst. Video technol.*, vol. 13, pp. 645-656, July 2003.
- [10] 남정학, 심동규 외 6명, "화소단위 예측에 의한 무손실 영상 부호화," *전자공학회 논문지*, 제 43권 SP편, 제 6호, pp. 97-104, 2006년 11월
- [11] 정광수, 이선오, 심동규, "H.264/AVC의 경계 세기 통계를 이용한 디지털 비디오에서의 객관적 화질 측정," *전자공학회 논문지*, 제 45권 SP편, 제 3호, pp. 64-73, 2008년 5월
- [12] 이제윤, 최웅일, 전병우, 석민수, "H.264의 가변 블록 움직임 보상을 위한 고속 움직임 벡터 탐색 및 모드 결정법," *전자공학회 논문지*, 제 40권 SP편, 제 4호, pp. 49-59, 2003년
- [13] 전혜민, 서정훈, 이영렬, "블록의 잔여 변환 계수를 이용한 적응적인 QP선택," *방송공학회논문지*, 제 14권, 제 2호, pp. 219-227, 2009년 3월
- [14] M. Horowitz, A. Joch, "H.264/AVC Baseline Profile Decoder Complexity Analysis," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 13, No. 7, pp. 704-716, July 2003.
- [15] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video technol.*, vol. 13, pp. 614-619, July 2003.
- [16] JM Reference Software Version 12.4, <http://iphome.hhi.de/suehring/tml/>
- [17] G. Bjontegaard, "Calculation of average psnr differences between rd-curves," *ITU-T SG16/Q.6 Doc. VCEG-M33*, Austin, Texas, USA, April 2001.
- [18] TK Tan, G. Sullivan, T. Wedi, "Recommended Simulation Conditions for Coding Efficiency Experiments Revision 3," *ITU-T SC16/Q.6, Doc. VCEG-A110*, Berlin, Germany, July 2008.
- [19] J. Ren and N. Kehtarnavaz, "Algorithmic Optimization for H.264 Deblocking Filter on Portable Devices," *ISCE 2007. IEEE International Symposium*, pp. 1-6, June 2007.
- [20] J. Lou, A. Jagmohan, Dake He, Ligang Lu, and M.T. Sun, "Statistical Analysis Based H.264 High Profile Deblocking Speedup," *ISCAS 2007. IEEE International Symposium*, pp. 3143-3146.

저 자 소 개



조 현 호(학생회원)
2008년 광운대학교 컴퓨터공학과
학사
2010년 광운대학교 컴퓨터공학과
석사
2010년~현재 광운대학교 컴퓨터
공학 박사과정.

<주관심분야 : 영상처리, 영상압축>



남 정 학(학생회원)
2006년 광운대학교 컴퓨터공학과
학사
2008년 광운대학교 컴퓨터공학과
석사
2008년~현재 광운대학교 컴퓨터
공학 박사과정

<주관심분야 : 영상압축, 멀티프로세서>



정 광 수(학생회원)
2008년 광운대학교 컴퓨터공학과
학사
2010년 광운대학교 컴퓨터공학과
석사
<주관심분야 : 영상신호처리, 영
상압축, 영상 화질측정>



심 동 규(정회원)
1999년 서강대학교 전자공학과
공학박사
1999년~2000년 (주) 현대 전자
2000년~2002년 (주) 바로 비전
2002년~2005년 Univ. of
Washington

2005년~현재 광운대학교 컴퓨터공학과 (부교수)
<주관심분야 : 영상신호처리, 영상압축, 컴퓨터비
전>



조 대 성(정회원)
1994년 서강대학교 전자공학과
학사
1996년 서강대학교 전자공학과
석사
1996년~2008년 삼성종합기술원
컴퓨팅랩 전문연구원

2008년~현재 삼성전자 DMC 연구소 멀티미디어
연구팀 수석연구원
<주관심분야 : 영상처리, 영상압축>



최 응 일(정회원)
2000년 성균관대학교 전기전자 및
컴퓨터공학부 학사
2002년 성균관대학교 전기전자 및
컴퓨터공학부 석사
2010년 성균관대학교 전기전자 및
컴퓨터공학부 박사

2006년~2008년 삼성종합기술원 컴퓨팅랩 전문
연구원
2008년~현재 삼성전자 DMC 연구소 멀티미디어
연구팀 책임연구원
<주관심분야 : 멀티미디어 프로세싱, 비디오코
딩>