논문 2007-44CI-6-4

H.264/AVC 비디오 코덱을 위한 효율적인 자료 재사용 디블록킹 필터 알고리즘

(An Efficient Data-reuse Deblocking Filter Algorithm for H.264/AVC)

이 형 표*, 이 용 석**

(Hyoung Pyo Lee and Yong Surk Lee)

요 약

H.264/AVC 표준은 복호된 영상의 블록간 경계면에서 발생하는 왜곡 및 불연속성을 보정하기 위하여 디블록킹 필터를 사용하여 더 나은 품질을 제공하였다. 하지만 이 과정에서 수많은 메모리 참조연산이 불가피하였으며, 이로 인해 전체 복호기의 처리 시간을 지연시키는 결과를 초래하였다. 본 논문에서는 이러한 디블록킹 필터의 처리 속도를 높이기 위한 자료 재사용 알고리즘을 제안한다. 자료 재사용을 위하여, 새로운 필터링 순서를 제안하여 메모리 참조를 감소시키고 디블록킹 필터의 처리 속도를 높인다. 제안된 알고리즘을 모델링하여 ARM ADS1.2에서 컴파일하고 ARM966E-S프로세서 시스템을 Armulator를 이용하여 시뮬레이션 하였다. 실험 결과, H.264/AVC 표준보다 매크로블록 당 실행 사이클, 메모리 참조 사이클에서 각각 58.45%, 57.93%의 성능 향상을 보였다.

Abstract

H.264/AVC provides better quality than other algorithms by using a deblocking filter to remove blocking distortion on block boundary of the decoded picture. However, this filtering process includes lots of memory accesses, which cause delay of overall decoding time. In this paper, we propose a data-reuse algorithm to speed up the process for the deblocking filter. To reuse the data, a new filtering order is suggested. By using this order, we reduce the memory access and accelerate the deblocking filter. The modeling of proposed algorithm is compiled under ARM ADS1.2 and simulated with Armulator. The results of the experiment compared with H.264/AVC standard are achieved on average 58.45% and 57.93% performance improvements at execution cycles and memory access cycles, respectively.

Keywords: deblocking filter, H.264/AVC, MPEG, video coding

I. 서 론

H.264/AVC는 기존의 MPEG-4 및 H.263 표준보다 약 두 배 더 높은 압축률과 개선된 품질을 제공하기 위해 JVT에 의해 제안되었다[1]. H.264/AVC의 주요한 특징으로는 향상된 움직임 예측 기법 (Motion estimation)의 도입, CAVLC(Content Adaptive Variable Length Coding) 및 CABAC(Content Adaptive Binary Arithmetic Coding)과 같은 효율적인 엔트로피 코딩, 디블록킹 필터의 향상 등이 있다[1, 8].

이 중, 디블록킹 필터는 복호된 영상의 블록의 경계면에서 발생하는 왜곡을 부드럽게 하기 위한 필터로서 블록 간 경계를 기준으로 좌우, 혹은 상하에 위치한 8개의 픽셀들의 관계, 경계면의 강도 및 양자화 계수 등을 고려하여 수행된다. 하지만, 이 필터링 작업은 수직 혹은 수평 경계면의 양쪽에 위치한 픽셀 값을 메모리로부터 읽고, 필터링 연산 후 다시 메모리에 저장함으로

* 학생회원, ** 평생회원, 연세대학교 전기전자공학과
(Department of Electrical and Electronic Engineering, Yonsei University)

※ 본 연구는 한국과학기술재단 특정기초연구(No. R01-2006-000-10156-0)지원으로 수행되었으며, IDEC (IC Design Education Center)에 의해 지원되는 EDA 툴이 사용되었습니다.
접수일자: 2007년5월16일, 수정완료일: 2007년11월4일

서 수많은 메모리 참조가 불가피하며 이로 인해 전체 복호기의 처리 시간이 지연되는 결과를 초래하게 된다.

이런 문제점을 개선하기 위해 메모리 참조를 줄일 수 있는 효율적인 메모리 구조 및 필터링 순서에 대한 연구들이 진행되었다. [3]은 경계면 단위가 아닌 블록 단위로 필터링하여 메모리로부터 읽어온 픽셀 값을 재사용 할 수 있는 필터링 순서를 제안하였으며, [4]는 수직 및 수평 경계면에 대해 교대로 필터링하여 자료 재사용의 효율을 높일 수 있는 필터링 순서를 제안하였다. 그러나 [3]과 [4]에서는 블록 단위의 로드가 필요하게 되므로 프로세서 기반의 범용 시스템에서는 캐시의 구조에 따라 이전에 사용한 픽셀 값이 캐시에서 지워질 수 있으며, 이 경우 다시 메모리로부터 픽셀 값을 읽어야 하므로 성능이 저하될 수 있다. 특히 [4]는 [3]보다 더 많은 픽셀 값을 저장하고 있어야 하므로, 프로세서 기반의 시스템에서는 자료 재사용에 대한 효율이 더욱 떨어질 수 있다. 따라서 한번 읽어 캐시에 저장된 픽셀 값은 최대한 많이 사용하는 것이 수행시간을 단축시키는데 도움이 되므로, 본 논문에서는 블록 단위가 아닌 라인 단위의 필터링을 제안하여 픽셀 값에 대한 재사용을 최대화 하며, 그 외에 필터링의 기준이 되는 경계면의 강도를 결정하는 부분에 대해서 자료를 재사용 할 수 있는 효율적인 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 본론에서는 H.264/AVC 표준의 디블록킹 필터 알고리즘과 [3]의 필터링 순서에 대해 알아보고, 제안하는 알고리즘을 설명한다. 실험에서는 H.264/AVC 표준과 [3]의 알고리즘, 그리고 본 논문에서 제안하는 알고리즘을 각각 모델링하여 시뮬레이션하고 그 결과를 비교한다. 마지막으로 결론에서는 실험 결과를 비교하여 분석한다.

Ⅱ. 본 론

1. 디블록킹 필터 알고리즘

디블록킹 필터는 H.264/AVC 표준의 주요한 특징 중에 하나로서, 복호 과정에서 역 양자화 및 역 변환에 의해 매크로블록의 경계면에 존재하는 블록 간의 왜곡 및 불연속성을 보정하기 위해 사용된다. 이는 인접한 블록 간의 경계면의 강도를 결정하는 단계, 그리고 결정된 경계면의 강도와 필터의 입력이 되는 픽셀 값의 변화를 이용하여 필터링 연산을 하는 단계로 이루어진다.

첫 번째 단계로, 경계면의 강도는 '0'에서 '4'까지의 값을 가질 수 있는데, 블록의 경계면이 매크로블록의

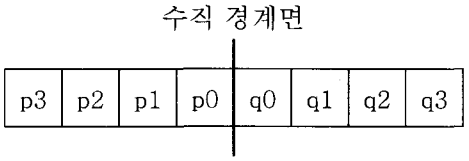

그림 1. 경계면을 기준으로 양쪽에 위치한 픽셀
Fig. 1. Samples in both side of block boundary.

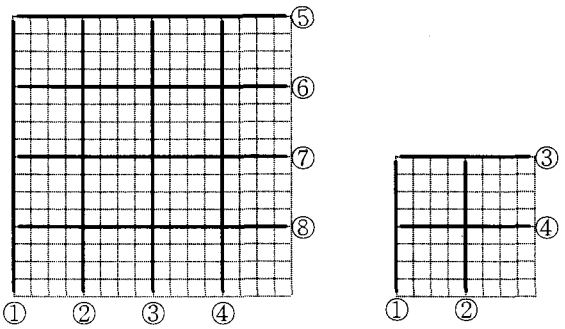

그림 2. H.264/AVC 표준의 디블록킹 필터 순서, (a) 휘도신호 매크로블록, (b) 색차신호 매크로블록
Fig. 2. Filtering order of H.264/AVC standard, (a) luminance MB, (b) chrominance MB.

경계인지의 여부, 필터의 입력이 되는 경계면의 좌우 또는 상하에 위치한 픽셀이 인트라 코딩된 픽셀인지의 여부, 그리고 사용되는 참조 영상 및 움직임 벡터의 동일 여부 등에 따라 결정된다.

디블록킹 필터의 두 번째 단계로, 결정된 경계면의 강도 및 경계면 양쪽에 위치한 픽셀들의 관계에 따라 적절한 3~5 탭 FIR 필터가 사용되는데, 경계면의 강도가 클수록 더 강한 필터가 사용되고, 경계면의 강도가 '0'일 경우에는 필터링하지 않는다. 필터링은 그림 1에서 보이는 바와 같이 경계면의 양쪽에 위치한 8개의 픽셀에서 이루어지며, 픽셀간의 크기 변화의 정도와 H.264/AVC 표준에서 정한 계수(α, β)간의 관계에 따라 필터의 세부동작이 결정된다[9].

필터링은 매크로블록 단위로 이루어지며, 그림 2에서 보는 바와 같이 먼저 블록 간의 수직 경계면에 대해서 왼쪽에서 오른쪽으로 수행되고, 그 다음에 수평 경계면에 대해서 위쪽에서 아래쪽으로 수행된다[1~2, 6]. 이 때, 하나의 매크로블록에서 수행되는 필터링의 총 회수는 그림 2에서 나타낸 바와 같이 16x16 휘도 신호 매크로블록에서 128회, 2개의 8x8 색차 신호 매크로블록에서 64회로 총 192회가 된다.

또한, 한 번의 필터링을 위해서 경계면의 양쪽에 위치한 8개의 픽셀을 메모리로부터 읽어야 하는데, 수직

경계면에 대해서는 8개의 픽셀이 메모리에 연속적으로 위치하고 있어서 한번에 4 바이트를 읽을 수 있지만, 수평 경계면에 대해서는 그렇지 않으므로 한번에 1 바이트밖에 읽을 수 없다. 따라서 하나의 매크로블록을 필터링하기 위해서 메모리에서 픽셀을 읽어오는 회수는 수직 경계면에 대해서 192회, 수평 경계면에 대해서 768회로 총 960회가 되며 이로 인해 디블록킹 필터의 성능 감소 및 소비 전력 증가 현상이 발생하게 된다.

2. 향상된 디블록킹 필터 알고리즘

H.264/AVC 의 메모리 참조에 대한 이러한 문제점을 개선하기 위해 [3]에서는 효율적인 필터링 순서와 메모리 구조를 이용하여 향상된 디블록킹 필터 알고리즘을 제안하였다.

[3]은 인접한 블록 단위로 필터링하는 그림 3과 같은 필터링 순서를 제안하여 자료를 재사용하였다. 즉, 그림 3의 ①에 대하여 필터링한 뒤, ①의 우측에 위치한 블록에 대한 자료를 메모리에 저장하지 않고 바로 ②에 대한 필터링에서 재사용함으로서 메모리에서 읽어오는 자료의 양을 감소시켰다. 또한, 두개의 2포트 SRAM을 이용한 메모리 인터리빙 구조를 적용하여 수평 경계면에 대한 필터링의 메모리 참조 문제를 개선하였다.

하지만 [3]과 같이 4x4 블록에 대해 동시에 필터링할 수 있는 구조에서는 블록만큼의 자료 재사용으로 성능을 향상 시킬 수 있지만, 프로세서 기반의 시스템에서는 동시에 연산할 수 있는 픽셀의 개수와 임시 데이터를 저장하는 레지스터의 개수가 제한적이므로, 자료의 재사용 측면에서도 일부 원하는 만큼의 효과를 얻기 힘들다.

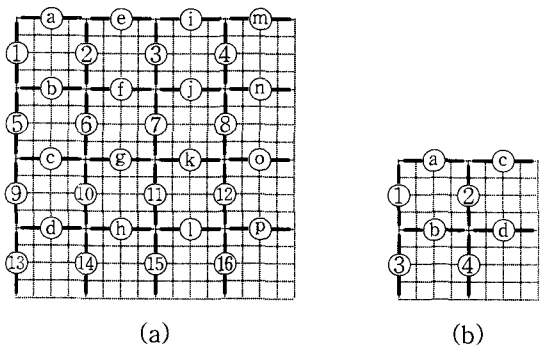

그림 3. 향상된 디블록킹 필터 순서, (a) 휘도신호 매크로블록, (b) 색차신호 매크로블록

Fig. 3. Advanced filtering order, (a) luminance MB, (b) chrominance MB.

3. 제안하는 알고리즘

가. 기준 프로세서 시스템

그림 4는 본 논문에서 제안하는 알고리즘을 적용하기 위한 기준 프로세서 시스템을 나타낸다. 이는 분리된 지역 명령어 메모리와 지역 데이터 메모리, 그리고 분리된 명령어 버스와 데이터 버스를 사용하며, 다중프로세서 환경을 지원하기 위하여 네트워크-온-칩(NoC: Network-on-Chip) 라우터를 포함한다.

프로세서는 5단 파이프라인으로 구성되어 있으며, 패치 단계에서는 명령어를 패치하고, 복호 단계에서 이를 복호 하여 실행 단계에서 연산을 수행한다. 메모리 단계에서는 메모리로부터 읽어오기 혹은 메모리에 쓰기 동작을 하고, 기록단계에서는 연산된 결과를 레지스터에 업데이트한다.

NoC 라우터는 다중프로세서 환경에서 다른 프로세서와의 연결고리 역할을 하며, 이전 프로세서에서 수신부를 통해 필터링할 영상 자료를 입력 받아 지역 데이터 메모리에 저장한 다음, 파이프라인 프로세서에서 필터링하고, 처리된 영상 자료를 다시 송신부를 통해 다음 프로세서로 넘겨주는 동작을 한다.

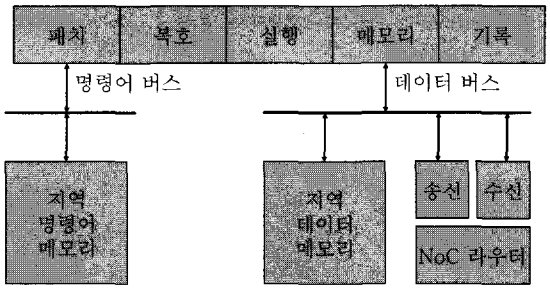

그림 4. 5단 파이프라인 시스템

Fig. 4. 5-stage pipeline system.

나. 제안하는 자료 재사용 알고리즘

[3]은 그림 3과 같이 블록 단위로 필터링하여 메모리에서 픽셀 값을 읽어오는 횟수를 감소시킴으로서 처리속도를 향상시켰다. 하지만, 기준 프로세서 환경에 [3]의 알고리즘을 적용할 경우 레지스터 파일의 개수가 적기 때문에 자료 재사용률이 현저히 감소되는 문제점이 있다. 따라서 본 논문에서는 블록 단위가 아닌 라인 단위로 필터링함으로서 자료 재사용률을 최대화한다.

그림 5는 제안하는 디블록킹 필터 순서의 흐름도를 나타내며, 이전 필터링의 결과를 다음 필터링에서 재사용하는 과정을 설명한다. 즉, 이전 필터링의 결과에서 경계면의 왼쪽 혹은 위쪽에 위치한 4개의 픽셀만 저장

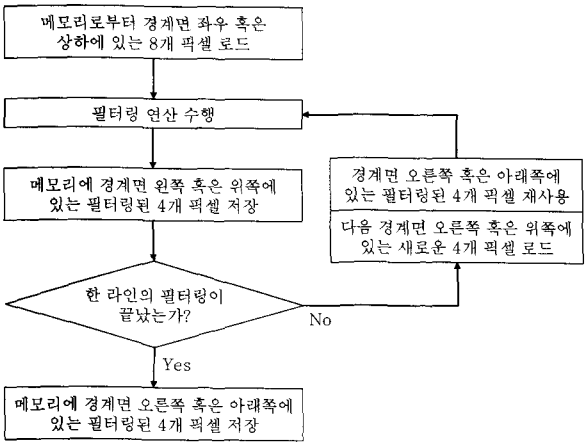

그림 5. 제안하는 디블록킹 필터 순서의 흐름도
Fig. 5. The flow chart of the proposed filtering order.

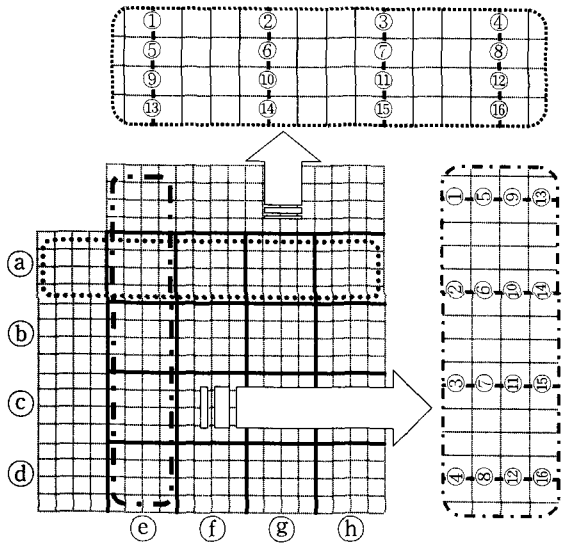

그림 6. 제안하는 디블록킹 필터 순서
Fig. 6. Proposed filtering order.

하고 오른쪽 혹은 아래쪽에 위치한 4개의 픽셀은 다음 필터링에서 재사용하며, 다음 필터링에서는 새로운 4개의 픽셀만 메모리로부터 읽어오는 과정을 나타낸다.

하나의 휘도신호 매크로블록에 대한 제안하는 필터링 순서를 그림 6에 자세히 나타내었다. 먼저 수직 경계면 ①의 양쪽에 위치한 픽셀 값을 메모리에서 읽어서 ①에 대해 필터링한 다음, ①의 왼쪽 픽셀 값만 메모리에 저장하고, 오른쪽 픽셀 값들은 다음에 수행되는 경계면 ②에 대한 필터링의 입력으로 재사용한다. 따라서 경계면 ②에 대한 필터링에서는 ②의 우측 픽셀 값만 메모리에서 읽어오게 됨으로써 메모리에서 읽어오는 횟수를 줄이게 된다. 이런 식으로 경계면 ④까지 필터링하고 다음 라인으로 넘어가게 되며, ⓐ에서 ⓓ까지 모든 수직 경계면에 대한 필터링이 끝나게 되면, 이어서 ⓔ에서 ⓗ까지 마찬가지 방법으로 수평 경계면에 대해 필터링한다. 결론적으로 자료 재사용률을 최대화하여 하나의 매크로블록에 대해 필터링하기 위해 각각의 픽셀들은 메모리로부터 한번만 읽어오게 되고 디블록킹 필터의 처리속도를 향상시키게 된다.

자료의 재사용은 블록의 경계면 강도를 결정하는 과정에도 이용될 수 있다. [3]은 필터링 순서와 효율적인 메모리 구조에 대해서 연구하여 그 처리속도를 향상시켰지만, 경계면의 강도를 결정하는 과정에 대해서는 고려하지 않았다. 본 논문에서는 경계면의 강도를 결정하는 데 사용되는 인자들이 블록 내에서 같은 값을 갖는다는 사실을 이용하여 이를 블록 단위로 수행하여 총 실행 횟수를 감소시키고, 필터링에서와 같은 방식으로 자료를 재사용하여 처리 속도를 향상시킨다.

III. 실 험

제안하는 알고리즘의 성능을 평가하기 위하여 디블록킹 필터를 수행함에 있어서 소요되는 실행 사이클 타임과 메모리를 참조하는데 소요되는 사이클 타임을 측정하였다. 컴파일러는 ARM ADS 1.2를 사용하였으며 ARM966E-S 프로세서와 64KB 명령어 캐시, 64KB 자료 캐시를 가정하였다. 이러한 시스템을 Armulator를 사용하여 리눅스 환경에서 시뮬레이션 하였다[5].

디블록킹 필터의 입력으로는 VCEG(Video Coding Expert Group)에서 권장하는 QCIF 및 CIF 영상을 H.264/AVC 표준으로 부호화한 파일이 사용되었다[7]. 표 1은 사용된 영상을 나타내며, 각각 다양한 양자화 계수와 프레임 생략을 적용하여 성능 비교의 수준을 높였다.

[1], [3] 그리고 제안하는 알고리즘을 이용하여 디블록킹 필터를 각각 C언어로 모델링하여 앞서 언급한 환

표 1. 실험의 입력으로 사용된 VCEG 권장 영상
Table 1. Input sequences recommended by the VCEG.

영상	영상 크기	프레임 수	양자화 계수	프레임 생략
Container	QCIF	300	20	2
Paris	CIF	300	16	1
Mobile	CIF	300	20	0
News	QCIF	300	24	2
Silent Voice	QCIF	300	20	1
Tempete	CIF	260	28	0
Foreman	QCIF	300	24	2

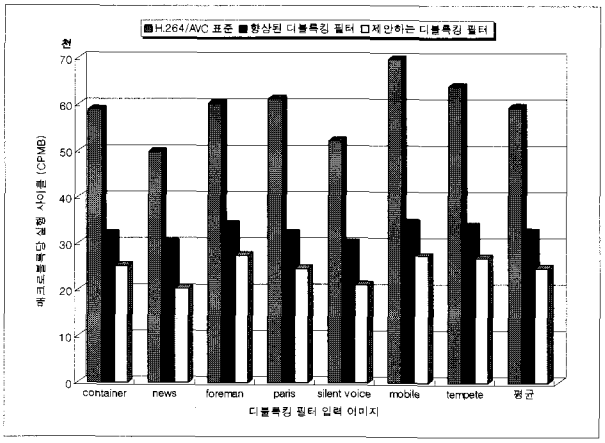

그림 7. 실행 사이클 비교
Fig. 7. Comparison of core-cycles.

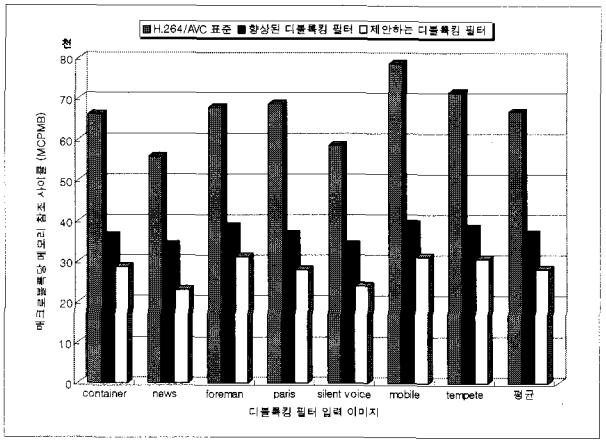

그림 8. 메모리 참조 사이클 비교
Fig. 8. Comparison of memory-cycles.

표 2. 제안하는 디블록킹 필터의 성능 비교
Table 2. Comparison of performance improvements.

알고리즘	CPMB(%)	MCPMB(%)
H.264/AVC 표준[1]	100	100
향상된 디블록킹필터[3]	54.63	55.09
제안하는 디블록킹필터	41.55	42.07

경에서 컴파일하고 시뮬레이션 하였다. 그리고 이를 토대로 매크로블록 당 필터링하는데 소요되는 실행 사이클(CPMB) 및 메모리 참조 사이클(MCPMB)등을 각각 그림 7과 그림 8에서 비교하였다.

CPMB와 MCPMB 두 가지 경우에 있어서 비슷한 경향의 성능 향상을 보였으며, 영상이 섬세하면서 다양한 색차 신호를 포함하고 전체적으로 움직임이 많이 존재하는 "Mobile" 에서 가장 많이 필터링함으로 인하여 약 61%로 가장 큰 성능 향상을 보였다. 하지만 이와는 반대로, 정지된 배경을 포함하는 "Paris" , "Silent Voice" 및 "News" 는 상대적으로 적게 필터링하였지만, 경계면을 결정하는 부분이 성능 향상에 영향을 주어 약 59%의 큰 성능 향상을 보였다.

제안하는 디블록킹 필터의 성능 향상을 비교하기 위해 표 1의 영상에 대한 실험 결과의 평균값을 H.264/AVC 표준을 기준으로 정규화하여 표 2에 나타내었다.

Ⅳ. 결 론

본 논문에서는 H.264/AVC 표준의 디블록킹 필터의 처리 속도를 향상시키기 위한 효율적인 자료 재사용 알고리즘을 제안하였다. 자료의 재사용률을 높이기 위한 필터링 순서를 제안하였으며, 경계면의 강도 결정을 블록 단위로 수행함으로서 처리속도를 향상시켰다. 성능 확인을 위해 VCEG에서 권장하는 7개의 영상을 입력으로 시뮬레이션 하여 [1]과 [3]에 비해 각각 CPMB에서 58.45%, 23.95%, MCPMB에서 57.93%, 23.63%의 성능 향상을 확인하였다.

H.264/AVC 표준의 처리속도를 개선하기 위한 각 기능별 가속화 엔진이 많이 연구되고 있으며, 이를 위해 ASIP(Application Specific Instruction-set Processor)을 통한 다중프로세서 환경이 많이 고려되고 있다. 본 논문에서 제안하는 디블록킹 필터는 이러한 구조에서 적합하며, ASIP으로 구현 시 더 좋은 성능 향상을 보일 것으로 기대된다.

참 고 문 헌

[1] JVT, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)," May 2003.
[2] L. E. G. Richardson, *"H.264 and MPEG-4 Video Compression,"* John Willey & Sons, pp. 184-187, Dec. 2003.
[3] Y.W. Huang, T.W. Chen, B.Y Hsieh, T.C. Wang, T.H. Chang, and L.G. Chen, "Architecture Design For Deblocking Filter In H.264/JVT/AVC," *International Conference on Multimedia and Expo,* pp. 693-696, Baltimore, Maryland, USA, July 2003.
[4] B. Sheng, W. Gao, and D. Wu, "An Implemented Architecture of Deblocking Filter for H.264/AVC," *International Conference on Image Processing,* pp. 665-668, Singapore, Oct. 2004.
[5] ARM Co., "Arm Developments Studio v1.2,"

http://www.arm.com
[6] K. Suhring, H.264/AVC software JM11.0, http://iphome.hhi.de/suehring/tml/, Nov. 2006.
[7] G. Sullivan and G. Bjontegaard, "Recommended Simulation Common Conditions for H.26L Coding Efficiency Experiments on Low-Resolution Progressive-Scan Source Material," ITU-T VCEG, Doc. VCEG-N81, Sep. 2001.
[8] T. Wiegand, G. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. on Circuits and Systems for Video Technology,* vol.13, no.7, pp.560-576, July 2003.
[9] P. List, A. Joch, J. Lainema, G. Bjontegaard and M. Karczewicz, "Adaptive Deblocking Filter," *IEEE Trans. on Circuits and Systems for Video Technology,* vol.13, no.7, pp.614-619, July 2003.

저 자 소 개

이 형 표(학생회원)
2001년 건국대학교 전자공학과 학사 졸업.
2001년~현재 삼성전자 DM총괄 선임연구원
2006년~현재 연세대학교 전기전자공학과 석사과정.
<주관심분야 : 영상신호처리, SoC 설계>

이 용 석(평생회원)
1973년 연세대학교 전자공학과 학사 졸업.
1977년 University of Michigan Electrical Engineering 석사 졸업.
1981년 University of Michigan Electrical Engineering 박사 졸업.
1993년~현재 연세대학교 전기전자공학과 교수.
<주관심분야 : 마이크로프로세서 설계, VLSI 설계, DSP 프로세서 설계, 고성능 연산기 설계>