논문 2004-41CI-1-2

멀티미디어 프로세서 구현에 사용되는 메모리를 줄이기 위한 저 복잡도의 영상 압축 알고리즘

(A Low-Complexity Image Compression Method Which Reduces Memories Used in Multimedia Processor Implementation)

정 수 운*, 김 이 랑*, 이 동 호**

(Su-Woon Jung, I-Rang Kim, and Dong-Ho Lee)

요 약

본 논문은 최근 많은 관심이 되고 있는 멀티미디어 프로세서 구현에 사용되는 메모리를 줄일 수 있는 성능이 우수하면서 하드웨어적으로 쉽게 구현이 가능한 영상 압축 알고리즘을 제안한다. 특히 고화질 영상을 처리하는 멀티미디어 프로세서는 영상 프레임을 저장하기 위하여 외부에 많은 양의 프레임 메모리를 사용하며, 또한 대부분의 프로세서에서 저 대역 필터와 같은 선형 필터를 구현하기 위하여 많은 양의 라인 메모리를 프로세서 안에 포함한다. 이러한 메모리 들은 멀티미디어 프로세서를 구현하는데 있어서 많은 비중을 차지하기 때문에, 만약 화질의 손상이 없으면서 이러한 메모리를 대폭 줄일 수 있다면 프로세서의 경쟁력을 높일 수가 있다. 기존의 JPEG과 같은 표준 압축 방법은 2차원 블록 단위로 처리하고 구현하기에 복잡하기 때문에 멀티미디어 프로세서에서 요구하는 래스터 스캔 입출력을 갖는 용도에는 적합하지가 않다. 본 논문에서는 래스터 스캔의 입출력을 위해 1x8 블록 단위로 처리하고 하드웨어적으로 쉽게 구현하고 압축 효율을 높이기 위해 Hadamard 변환을 이용하고, 변환된 계수의 특성을 분석하여 그에 따라 적응적으로 thresholding을 적용한 후 양자화를 하였다. 모의실험을 통해 메모리를 반으로 줄였을 때 기존의 압축 방법과 성능을 비교하였으며, 하드웨어의 구현을 통해 멀티미디어 프로세서를 구현하는데 있어서 어느 정도 경쟁력을 높일 수 있는 지를 분석하였다.

Abstract

This paper presents an efficient image compression method for memory reduction in multimedia processor which can be simply implemented in hardware andprovides high performance. The multimedia processor, which includes processing of high-resolution images and videos, requires large memories: they are external frame memories to store frames and internal line memories for implementing some linear filters.If we can reduce those memories by adopting a simple compression method in multimedia processor, it will strengthen its cost competitiveness. There exist many standards for efficiently compressing images and videos. However, those standards are too complex for our purpose and most of them are 2-D block-based methods, which do not support raster scanned input and output. In this paper, we propose a low-complexity compression method which has good performance, can be implemented with simple hardware logic, and supports raster scan. We have adopted 1x8 Hadamard transform for simple implementation in hardware and compression efficiency. After analyzing the coefficients, we applied an adaptive thresholding and quantization. We provide some simulation results to analyze its performance and compare with the existing methods. We also provide its hardware implementation results and discuss about cost reduction effects when applied in implementing a multimedia processor.

Keywords: Multimedia, Image Compression, JPEG, DTV, 프레임 메모리

* 학생회원 **정회원, 한양대학교 전자컴퓨터공학부
(School of Electrical and Computer Engineering, Hanyang University)

※ 본 연구는 한국과학재단 목적기초연구(과제번호 R01-2003-000-10758-0) 지원으로 수행되었음.
접수일자 : 2003년 6월 17일, 수정완료일 : 2004년 1월 7일

Ⅰ. 서 론

방송, 통신, 디스플레이 등의 기술의 발전에 힘입어 고해상도의 영상 처리를 포함하는 멀티미디어 프로세서의 개발이 세계적으로 많은 주목을 받고 있다. 고해상도의 영상을 처리하기 위해서는 단위 시간당 많은 데이터를 처리해야 하는 문제점도 있지만 많은 메모리를 필요로 한다. 특히, 영상의 프레임을 저장하기 위해서는 프로세서 외부에 프레임 메모리를 사용해야 되고, 대부분의 멀티미디어 프로세서 내부에는 저역 통과 필터 같은 선형 필터를 구현하기 위하여 많은 라인 메모리를 포함해야 하는 문제점이 있다. 저해상도의 영상에서는 이러한 메모리가 차지하는 비중이 크지 않을 수도 있지만 고해상도의 영상을 처리하기 위해서는 요구되는 방대한 메모리가 구현하는데 가장 큰 문제점이 되고 있다. 따라서 이러한 멀티미디어 프로세서를 구현하는 데 있어서 요구되는 메모리를 대폭 줄일 수 있다면 프로세서의 가격 경쟁력을 높일 수가 있을 것이다. 즉 간단하고 효율적인 영상 압축 알고리즘을 프로세서 내부에 포함시켜서 내부 메모리를 줄인다면 칩의 면적을 줄임으로서 원가를 낮출 수가 있고, 내부의 이러한 압축 기법을 이용하여 외부 메모리 사용을 위한 프로세서의 핀 수를 줄임으로서 칩의 크기를 줄일 수 있을 뿐만 아니라 시스템을 구현하는데 요구되는 외부 메모리의 용량을 줄일 수 가 있기 때문이다.

이러한 영상 압축 알고리즘에는 몇 가지 제약 조건이 따른다. 첫째로 압축을 시도했을 때 복원된 영상이 원 영상과 비교하여 화질의 열화가 없어야 한다. 둘째로는 복잡도가 낮아 하드웨어 로직으로 쉽게 구현할 수가 있어야 하고 출력이 나오기 까지 지연이 적어야 한다. 마지막으로 영상의 입출력이 래스터 스캔 방식으로 되기 위해 일 차원 단위로 압축이 되어야 한다. 이러한 정지 영상을 압축하는 대표적인 방법으로 JPEG을 들 수가 있는 데[1], 성능은 매우 우수하나 멀티미디어 프로세서에 상기의 목적으로 적용하기에는 많은 문제점을 갖고 있다. 첫째로 구현하는데 있어서 복잡도가 너무 크고 출력이 나오기 까지 많은 지연이 따른다. 둘째로는 2 차원 블록 단위로 압축을 하기 때문에 래스터 스캔의 입출력을 지원하지 못하는 문제점이 있다. 이 외에도 ADPCM 압축 방법[2][3] 을 들 수가 있는데, 이러한 방법도 성능은 우수한 편이나 복잡도가 높고 많은 지연을 갖기 때문에 적용하기에는 적절하지 못하다고 할 수 있다.

본 논문에서는 래스터 스캔의 입출력을 위해 1x8 블록 단위로 처리하고 하드웨어적으로 간단하게 구현하는 효율적인 영상 압축 알고리즘을 제안한다. 제안하는 알고리즘은 별도로 압축 율을 제어할 필요가 없으며, 압축 효율을 높이기 위해 Hadamard 변환을 이용하고, 변환된 계수의 특성을 분석하여 그에 따라 적응적으로 thresholding을 적용한 후 양자화를 하였다. 또한 모의실험을 통해 메모리를 반으로 줄였을 때 기존의 압축 방법과 비교하여 성능의 우수성을 제시하였고, 하드웨어의 구현을 통해 멀티미디어 프로세서를 구현하는데 있어서 어느 정도 가격 경쟁력에 기여할 수 있는 가를 분석하였다.

본 논문의 전개는 다음과 같다. 먼저 제Ⅱ장에서는 기존 방법들의 문제점에 관하여 논하고, 제Ⅲ장에서는 제안하는 압축 알고리즘에 관하여 구체적으로 설명한다. 제Ⅳ장에서는 모의실험을 통해 제안하는 알고리즘의 성능을 비교 및 분석하였다. 제Ⅴ장에서는 하드웨어 구현 결과와 이를 통해 멀티미디어 프로세서의 경쟁력에 미치는 효과를 분석하였고, 제Ⅵ장에서 결론을 맺는다.

Ⅱ. 기존 방법들의 문제점

화면 간의 상관성을 이용하지 않고 입력되는 신호를 실시간으로 압축하기 위한 가능한 대표적인 방법으로는 JPEG(Joint Photographic Expert Group)과ADPCM (Adaptive Differential Pulse Code Modul- ation)을 들 수 있다[1]-[3].

JPEG은 컬러 정지 영상의 압축 표준으로 화질의 열화가 없는 무 손실 방식과 화질의 열화를 허용하면서 높은 압축 율이 가능한 손실 방식이 있다. 손실 방식 중에서도 가장 널리 적용되는 방식은 baseline 방식이다. 무 손실 방식은 일정한 압축 율을 보장할 수 없고 대부분의 영상에서 2배로 압축하는 것이 어렵기 때문에 JPEG baseline에대해서만 비교를 하였다. JPEG baseline에서 입력 영상은 8x8 블록 단위의 DCT(Discrete Cosine Transform)를 거쳐 양자화를 수행한다. 양자화 된 계수는 허프만 테이블을 이용한 VLC(Variable Length Coding)를 거쳐 출력되는데, 압축 정도는 양자화를 통해 조절이 된다. 이러한JPEG baseline의 문제점은 첫째로 8x8 블록 단위로 처리되기 때문에 래스터 스캔의 입력을 처리하기 위해서는 블록의 크기에 비례하는 라인 메모리를 포함해

야만 한다. 다른 문제점은 양자화된 계수에 대해 VLC를 적용하기 때문에 원하는 비트의 양으로 압축하기가 매우 어렵고 이를 해결하기가 쉽지 않은 문제점이 있다. 또한 JPEG baseline을 하드웨어로 구현하기에는 상당한 로직과 메모리를 필요로 하기 때문에 본 논문의 목적에는 적절하지 않다고 할 수 있다.

다음으로 생각할 수 있는 압축 방법으로는 ADPCM을 들 수 있다. ADPCM은 JPEG baseline과 같은 표준 방법은 아니고 입력영상에 대해 일정한 블록 단위로 DPCM(Differential Pulse Code Modulation)을 수행하여 비선형적인 최적 양자화기를 적용하는 방법이다. ADPCM은 매 프레임 마다 적응적인 양자화기를 설계하여 적용하기 때문에 성능은 우수한 편이나 적용하기에는 몇 가지 문제점을 갖고있다. 특히 현재 입력되는 프레임에 대한 양자화기를 설계하기 위해서는 프레임에 대한 분산 값을 계산해야 하는데, 이를 위해서는 별도의 프레임 메모리를 필요로 하고 또한 적응적인 양자화기를 설계하기 위해서는 코드북을 저장하는 별도의 메모리를 필요로 한다. 이러한 복잡도로 인해 ADPCM도 적용하기에는 어려움이 있다.

또한 MPEG-2와 같은 비디오 디코더를 구현하는데 있어서 외부의 프레임 메모리를 줄이기 위한 방법으로 DCT 영역에서 보간 방법을 이용하는 방법들이 발표되었다[4]-[10]. 이러한 연구들은 MPEG-2 비디오 디코더 구현에만 적용이 가능하다는 문제점이 있다.

Ⅲ. 제안하는 영상 압축 알고리즘

그림 1에는 본 논문에서 제안하는 영상 압축 알고리즘의 전체 블록도를 나타내었다. 데이터의 입출력이 래스터 스캔을 지원하기 위하여 압축하는 단위는 1x8 블록(8 bytes)으로 처리하였다. 다양한 블록 크기에 대해 실험을 한 결과 1x8 크기의 블록이 성능이 가장 우수함을 보였기 때문이다. 이러한 압축 알고리즘은 프레임 메모리를 반으로 줄이는 것을 목표로 하였으며, 이는 크게 3 단계의 과정을 거친다. 먼저 입력되는 영상에 대해 압축 효율성을 높이기 위해 1x8 블록 단위로 Hadamard 변환을 통해 계수 영역으로 변환을 한다. 다양한 영상에 대해 각각의 계수의 통계 분포를 조사하여 DC 성분을 제외하고 나머지 계수에 대해 대부분의 값이 존재하는 영역으로 각각의 계수들에 대해 적합한 thresholding을 적용하고 각 계수에 대해 독립적으로 양자화를 하게 된다.

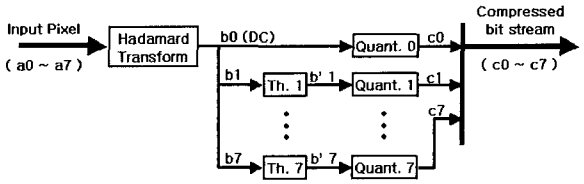


그림 1. 제안하는 영상 압축 알고리즘의 전체 구성도
Fig. 1. Overall block diagram of the proposed image compression method

1. Hadamard 변환

공간 영역에서 존재하는 상관성을 줄여 압축의 효율성을 높이기 위한 방법으로 변환이 사용되는데, 일반적으로 DCT(Discrete Cosine Transform)가 매우 우수한 성능을 보이며 널리 사용된다. 하지만 본 논문에서는 하드웨어로 쉽게 구현이 가능한 저 복잡도의 알고리즘을 위하여 DCT 대신에 하드웨어 구현이 용이한 DHT(Discrete Hadamard Transform)을 채택하였다. 표 1에는 1x8 블록을 DCT와 DHT로 구현했을 때 요구되는 연산량을 나타내었다. DCT는 덧셈기 외에도 많은 양의 곱셈기가 요구되는 반면에 DHT는 덧셈기 만으로 구현이 가능하다는 것을 알 수 있다. 이러한 곱셈기는 하드웨어로 구현하는데 많은 로직을 차지하기 때문에 효율적이지 못하다.

표 1. 1x8 블록에 대한 DCT와 DHT의 연산량 비교
Table 1. Complexity comparison of DCT and DHT for 1x8 block

		Multiplier	Shifter	Adder	Subtractor
DCT	Forward	57	64	56	0
	Inverse	57	64	56	0
DHT	Forward	d0	0	29	28
	Inverse	0	0	29	28

2. Thresholding 및 양자화

Hadamard 변환을 한 후에 곧 바로 양자화를 적용할 경우 많은 화질의 열화를 피할 수가 없다. 특히 2배로 압축을 하기 위해 AC 계수에 적은 비트를 할 당할 경우 발생하는 손실은 복원된 영상에서 심각한 블록킹 현상으로 나타난다. 그림 2는 352x240 의 SD 급 해상도를 갖는 "Football" 영상에 대한 각 계수들의 분포를 나타낸다. 그림에서 보듯이 DC 계수를 제외한 나머지 계수들의 분포는 특정한 대역에서 Laplacian 분포를 갖는 것을 알 수 있다. 본 논문에서는 양자화를 통해 발생하는 손실을 최소화하기 위해 양자화 전에 이러한 분포를 근거로 하여

결정된 임계값(threshold)을 이용하여 식 (1)과 같이 thresholding을 적용하였다. 이 식에서 'threshold'는 다음의 방법을 통해 결정된 임계값을, 'bi'는 i번째 Hadamard 계수값을 그리고 'b'i' 는 thresholding 처리된 i번째 Hadamard 계수 값을 나타낸다. 먼저 한 영상에서 계수 각각에 대한 히스토그램을 구하고, DC 를 제외한 나머지 계수 각각에서 히스토그램 값이 0이 되는 상한과 하한값에 해당하는 임계값을 산출하였다. 다양한 해상도를 갖는 1000개의 영상에 대해서 같은 방법을 반복, 확률이 가장 높은 임계값을 결정하였다. 이렇게 결정된 임계값을 기준으로 하여 임계값 밖의 영역은 버리고 임계값 안의 영역에 대해서만 양자화하면 양자화 손실을 효과적으로 줄일 수 있다. 실제 영상에 있어서 계수값의 분포는 거의 유사성을 보인다. 따라서 이렇게 결정된 임계값은 모든 영상에서 동일하게 적용되며, 이때 사용되는 임계값 들은 다양한 테스트 영상에 대해 복원 후 영상의 손실이 최소화 됨을 확인하였다.

그림 3에는 입력되는 영상에 대해 Hadamard 변환,

$$b'i\ (i\ th\ Hadamard\ coeff.) = \begin{cases} -1 \times threshold, & bi < (-1 \times threshold) \\ threshold, & bi > threshold \\ bi, & otherwise \end{cases} \quad (1)$$

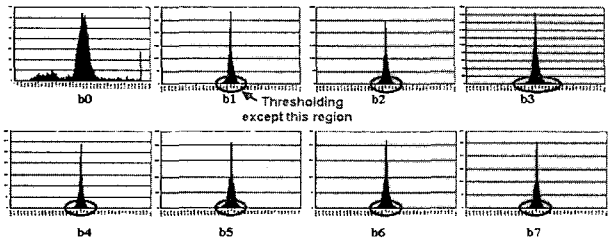


그림 2. DHT 계수들의 분포
Fig. 2. Distribution of DHT Coefficients

thresholding 및 양자화 되는 구체적인 내용을 도시적으로 나타내었다. 그림에서와 같이 DC 성분을 제외한 나머지 AC 계수들에 대해 일정한 범위를 갖는 값으로 thresholding을 하였다. 예를 들어 b5에 해당하는 AC 계수는 11 비트로 나타내는 범위를 thresholding을 통해 8 비트로 줄여서 나타내었다. 이렇게 thresholding된 계수 들에 대해 정확하게 2배로 압축이 되도록 양자화를 거치는데 그림에는 각 계수에 대해 양자화를 통해 할당되는 비트 수를 나타내었다. 이때 각 계수에 적용되는 양자화 스텝 크기는 다양한 테스트 영상에 대한 실험을 통해 에러가 최소화 되는 값으로 결정을 하였다. 이러한 간단한 양자화기를 통해 매 프레임 마다 적응적인 양자화기를 설계하여 적용하는 것에 비해 성능은 약간 못 미치지만 구현에 필요한 복잡도를 줄이고 압축과 복원 과정에 요구되는 지연을 최소화 시킬 수가 있다. 또한 별도의 압축 율을 조절하는 기능이 없어도 되고 1x8 블록 단위로 항상 2배로 압축하는 것이 보장되는 장점이 있다.

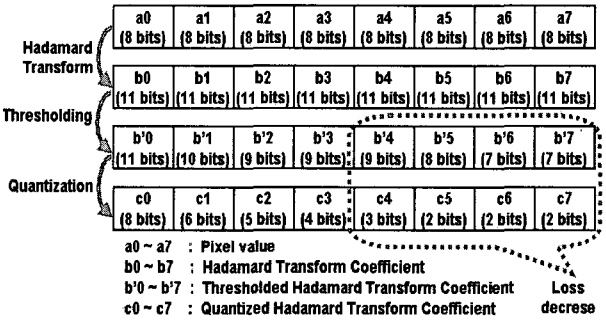


그림 3. Thresholding과 양자화 방법
Fig. 3. Detail method for thresholding and quantization

표 2. 제안하는 방법으로 압축했을 때 다양한 영상에 대한 성능 및 압축률 비교
Table 2. Performance comparison for test images

Reso-lution	High-freq.	Test image	Proposed algorithm	JPEG baseline	ADPCM algorithm	1X32 sub_block ADPCM
SD	Low	Football (704x480)	Performance 39.27 dB Compression ratio2 : 1	44.80 dB 2.04 : 1	42.41 dB 1.77 : 1	41.74 dB 1.86 : 1
		Susi (704x480)	45.77 dB 2 : 1	48.51 dB 2.19 : 1	46.42 dB 1.77 : 1	47.10 dB 1.86 : 1
		Ballet (720x480)	42.44 dB 2 : 1	45.14 dB 1.89 : 1	47.29 dB 1.77 : 1	47.29 dB 1.86 : 1
	High	Flower arden (704x480)	40.91 dB 2 : 1	45.44 dB 1.96 : 1	37.99 dB 1.77 : 1	37.13 dB 1.86 : 1
		Table Tennis (704x480)	39.71 dB 2 : 1	43.93 dB 2.01 : 1	41.16 dB 1.77 : 1	40.43 dB 1.86 : 1
HD	Low	Boxing (1920x1088)	48.39 dB 2 : 1	48.98 dB 1.99 : 1	45.26 dB 1.84 : 1	44.52 dB 1.86 : 1
	High	Quiz (1920x1080)	42.11 dB 2 : 1	45.36 dB 1.98 : 1	43.48 dB 1.84 : 1	42.78dB 1.86 : 1

IV. 모의 실험 결과

제안된 알고리즘의 성능을 분석하기 위해 다양한 테스트 영상을 이용하여 주관적인 평가와 객관적인 평가를 수행 하였다. 그림 4에는 HD급 해상도의 "boxing" 영상에 대해 처리하는 블록 단위를 1x8, 1x16, 1x32를 적용했을 때 복원한 영상과 원래 영상과의 PSNR을 나타내었다. 이때 thresholding과 양자화는 블록의 크기에 따라 같은 기준을 적용하여 임계치와 양자화 레벨을 구하여 2배로 압축했을 때의 결과이다. 그림에서와 같이 블록의 크기가 커질수록 성능은 오히려 떨어지고 블록의 크기가 1x8 인 경우가 최적의 크기임을 알 수 있다. 그림 5에는 다른 값으로 thresholding을 적용한 경우에 대한 출력 영상과 PSNR을 나타내었다. 본 논문에서 실험을 통해 결정한 threshold 값보다 각 AC 계수에 대해 1 비트씩을 늘이는 경우와 줄이는 경우에 대해 비교를 하였다. 그림에서와 같이 제안하는 방법이 다른 threshold를 적용하는 경우에 비해 주관적인 화질과 PSNR이 우수함을 알 수 있다.

표 2에는 다양한 테스트 영상에 대해 제안하는 방법을 이용하여 복원한 영상의 평균 PSNR과 압축률을 나타내었다. 또한 기존의 JPEG baseline과 ADPCM, 그리고 래스터 스캔 입력을 고려하여 1-D 로 처리되는 1x32 sub-block ADPCM 을 적용하여 PSNR과 압축률을 비교하였다. 표에서와 같이 SD급 해상도의 영상에 대해서는 39 dB 이상의 성능을, HD급 해상도의 영상에 대해서는 42 dB 이상의 성능을 보이는 것을 알 수 있다. JPEG baseline을 적용하는 경우에는 제안하는 방법에 비해 평균적으로 3-4 dB 이상 높은 성능을 얻을 수 있지만 래스터 스캔의 입력에 적용하는 것이 어렵고 구현에 요구되는 복잡도가 높기 때문에 본 논문에서 적용하려는 용도에는 적절하지 못하다. ADPCM의 경우에는 비슷한 성능을 보이지만 구현 상의 복잡도를 비교하면 제안하는 방법의 우수성을 확인할 수 있다.

그림 6에는 704x480 의 SD급 해상도의 "flower garden" 영상에 대해 thresholding을 생략한 경우와 적용한 경우에 대해 복원한 영상을 비교하였다. thresholding을 적용하지 않고 직접 양자화를 적용한 경우에는 심각한 화질의 열화가 있는 반면에 제안하는 thresholding을 적용한 경우에는 원래 영상과의 차이가 거의 없음을 알 수 있다. 그림 7과 8에는 HD급 해상도의 영상에 대해 원래 영상과 제안하는 방법을 적용하여 복원한 영상의 부분을 확대하여 비교하였다. 역시 원래 영상과의 차이가 거의 없음을 알 수 있다. 따라서 본 논문에서 제안하는 방법을 멀티미디어 프로세서에 적용하였을 때 메모리 크기는 반으로 줄이면서 화질에 있어서는 원래 영상과 큰 차이가 없음을 알 수 있다.

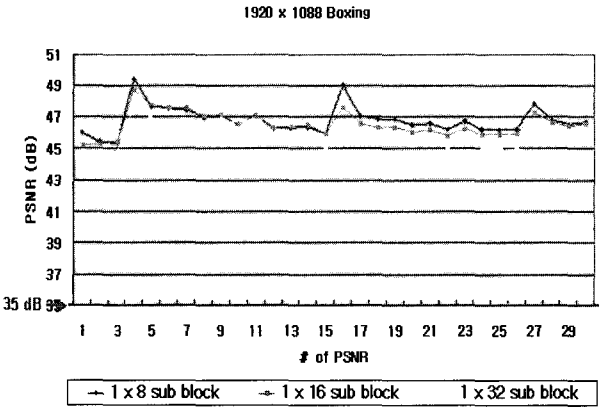


그림 4. 다른 블록 크기로 압축했을 때의 PSNR 비교
Fig. 4. Performance comparison when compressed with different block size

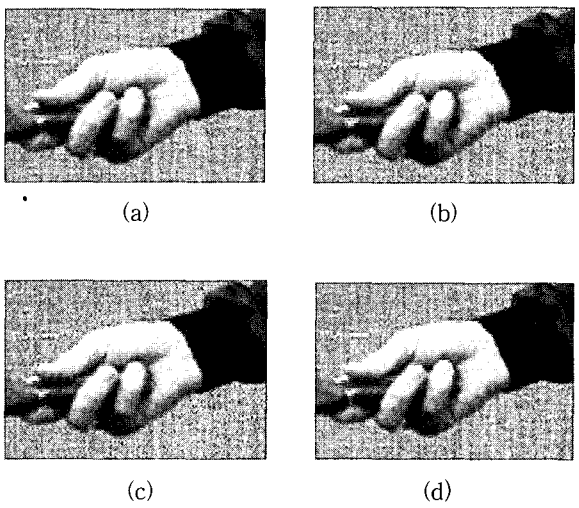


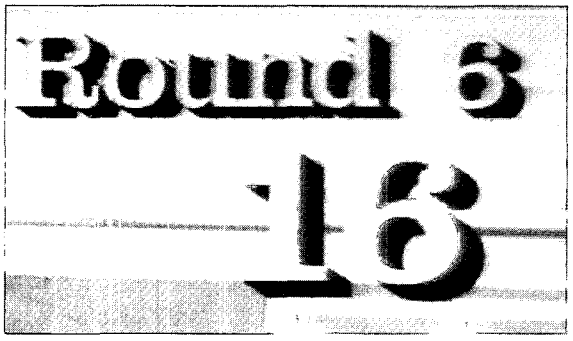
그림 5. threshold을 다르게 적용한 경우에 대한 비교 : (a) 원래 영상, (b) 1비트 추가한 threshold (35.88dB), (c) 1비트 줄인 threshold (34.58dB), (d) 제안하는 threshold (36.77dB)

Fig. 5. Original and reconstructed images when applied different threshold values: (a) original image, (b) 1 bit added threshold value (35.88dB), (c) 1 bit reduced threshold value (34.58dB), (d) the proposed threshold value (36.77dB)

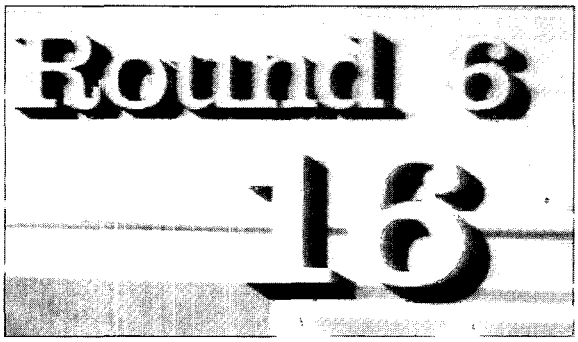
(a) (b) (c)

그림 6. "table tennis" 영상에 대한 모의실험 결과: (a) 원래 영상, (b) thresholding을 적용하지 않은 영상, (c) thresholding을 적용한 영상

Fig. 6. Original and reconstructed images: (a) original image, (b) when applied non thresholding, (c) when applied thresholding



(a)



(b)

그림 7. HD 급 "boxing" 영상에 대한 모의실험 결과: (a) 압축하기 전의 원래 영상, (b) 제안하는 방법으로 압축한 영상

Fig. 7. Simulation results for HD resolution image: (a) original image and (b) reconstructed image using the proposed method

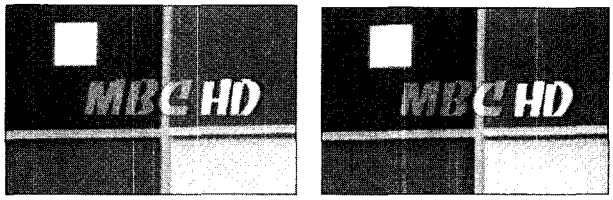


그림 8. HD 급 "quiz" 영상에 대한 모의실험 결과: (a)(c) 압축하기 전의 원래 영상, (b)(d) 제안하는 방법으로 압축한 영상,

Fig. 8. Simulation results for HD resolution image: (a)(c) original image, (b)(d) reconstructed image using the proposed method

V. 하드웨어 구현 결과와 효율성 분석

그림 9에는 제안하는 압축 방법을 하드웨어로 구현하기 위한 엔코더와 디코더의 구조를 나타내었다.

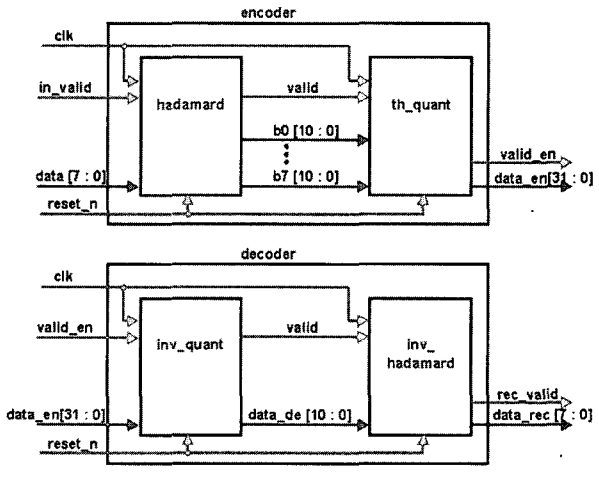


그림 9. 제안하는 압축 방법의 하드웨어 구현 구조

Fig. 9. Architecture for hardware implementation of the proposed compression method

엔코더는 8 바이트 단위의 영상과 시스템 클럭 그리고 데이터 유효 신호를 입력 받아 Hadamard 변환과 thresholding 및 양자화를 거쳐 2배로 압축이 된 4 바이트 데이터를 출력한다. 디코더는 4 바이트 압축된 데이터를 입력 받아 엔코더에 입력된 형태의 영상 신호를 복원하여 출력하게 된다. 이때 하드웨어를 구현하기 위하여 사용된 언어는 국제 표준인VHDL(Very high- speed integrated circuit Hardware Description Language)을 사용하였으며 동작 및 타이밍 모의실험은 Mentor 사의 Modelsim SE 5.6e 버전을 이용하였다. C 언어로 모델링하여 얻은 결과와 동작 및 타이밍 시뮬레이션을 통해 얻은 결과를 픽셀 단위로 비교하여 검증 하였다. 그림 10은 704x480 "Susi" 영상에 대해서 제안된 방법의 C 모델링 결과와 Modelsim을 이용한 VHDL모의실험 결과를 나타내었다. 실제로 각각의 화소 값을 비교했을 때 한치의 오차도 없는 결과를 얻었다.

표 3에는 하이닉스사의 0.35㎛ 표준 셀 공정을 이용하여 합성한 결과를 나타내었다. 표에서와 같이 제안하는 압축 방법을 하드웨어로 구현하는 데 요구되는 플립플럽의 수는 엔코더의 경우에 약 150이고 디코더의 경우에는 약 110개 정도가 요구되고 엔코더에서 11클럭, 디코더에서 12클럭의 총 23 클럭이 소요된다. 일반적으로 수만 개의 플립플럽을 요구하는 JPEG baseline를 구현하는데 요구되는 하드웨어 로직에 비하면 매우 단순하게 구현할 수 있음을 알 수 있다.

그림 11에는 외부 프레임 메모리를 줄이기 위해 실제 멀티미디어 프로세서 안에 제안하는 압축 방법을 하드웨어로 구현하여 포함시켰을 때의 구조와 인터페이스를 나타내었다. 이때 멀티미디어 프로세서 안에서 비디오 신호는 클럭과 함께 휘도와 색차 신호가 별도로 입력되는 것을 가정하여 설계하였고 기존의 외부 메모리 인터페이스 앞에 "Encoder" 와 "Decoder" 블록으로 구성된 "Top" 블록 단일 IP 처럼 추가함으로써 간단히 외부 프레임 메모리를 반으로 줄일 수가 있다.

(a)

(b)

그림 10. "Susi" 영상을 이용한 출력 영상: (a) C 모델링 결과, (b) Modelsim을 이용한 VHDL 시뮬레이션 결과

Fig. 10. Results comparison: (a) result of C modeling, (b) result of VHDL simulation using Modelsim

표 3. 설계한 압축 알고리즘의 하드웨에 합성 결과

Table 3. Hardware synthesis results of the proposed method

Block		Multi-plier	Shifter	adder	Acc-umul-ater	coun-ter	D-FF
Encoder	Hadamard	0	0	1	7	1	12
	Th_quant	0	0	12	0	0	125
Decoder	Inv_ hadamard	0	0	1	7	2	42
	Inv_quant	0	11	4	0	1	45
Total		0	11	18	14	4	224

예를 들어 1920x1080 해상도의 영상을 저장하는 경우에는 멀티미디어 프로세서 안에 앞에서 설명한 간단한 하드웨어 로직을 포함시킴으로써 한 프레임을 저장하기 위해 사용되는 메모리를 64Mbit에서 32Mbit로 줄이는 효과를 얻을 수가 있다. 즉 이러한 멀티미디어 프로세서를 이용하여 시스템을 구현 할 경우, 적용하지 않는 경우에 비해 사용되는 메모리 용량을 반으로 줄일 수가 있다. 메모리를 줄이는 효과 이외에도 메모리에 저장되는 비디오 데이터의 대역폭이 반으로 줄기 때문에 요구되는 외부 메모리의 엑세스 시간도 반으로 줄일 수 있고 요구되는 데이터 버스의 폭도 반으로 줄일 수가 있다. 일반적인 SDRAM을 이용하여 HD급 해상도의 영상을 저장할 경우에는 보통 64 비트의 데이터 버스를 이용하는데 이러한 경우에는 32 비트로 줄일 수 있기 때문에 멀티미디어 프로세서 칩의 핀 수를

32개 정도를 줄일 수가 있다. 이러한 경우에는 칩 패키지의 크기를 줄일 수 있기 때문에 칩의 단가를 줄이는 효과를 얻을 수가 있다.

멀티미디어 프로세서를 구현하는 데 있어서 프레임을 저장하기 위한 외부 메모리 이외에도 칩 내부에 선형 필터 등을 구현하기 위하여 내부에 라인 메모리를 사용하는 경우가 보통이다. 이러한 경우 사용되는 내부 메모리가 전체 프로세서의 상당 부분을 차지하고 메모리를 많이 사용할수록 이로 인해 칩을 구현하는 데 많은 어려움이 따른다. 그림 12에는 제안하는 압축 방법을 적용하여 일반적인 FIR 필터를 구현하는 구조를 나타내었다. 표 4에는 이러한 구조를 적용하였을 때 절감되는 내부 메모리의 크기를 나타내었다. 괄호 안에는 압축되는 비율을 나타내었다. 이때 1 비트를 저장하는 내부 메모리의 기본 단위가 플립플럽 1개와 복잡도가 같다고 가정을 하였으며 컬러 영상을 저장하는 경우를 고려하였다. 표에서와 같이 영상의 해상도가 낮은 경우에는 큰 이득이 없지만 영상의 해상도가 높을수록, 그리고 사용되는 라인 메모리가 많을수록 사용되는 내부 메모리의 크기를 거의 반 가까이 줄일 수 있음을 알 수 있다. 이렇게 내부 메모리를 줄임으로써 칩을 구현하는데 요구되는 면적의 크기를 대폭 줄임으로써 칩의 단가를 줄여 칩의 경쟁력을 높일 수가 있다.

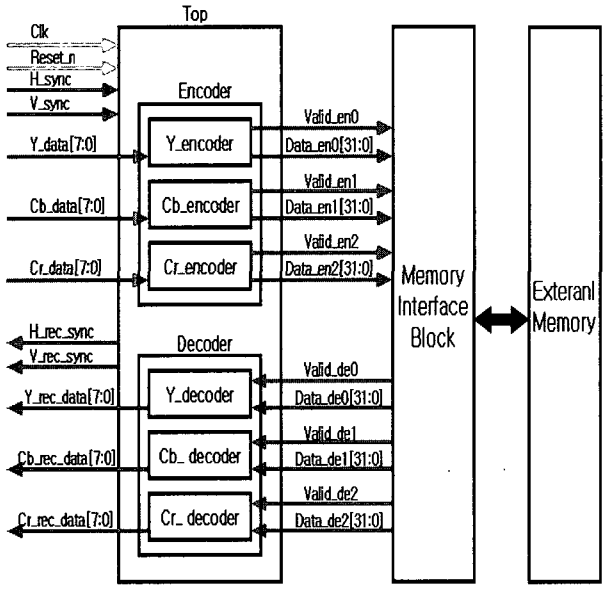


그림 11. 외부 프레임 메모리와 칩 내부에서의 제안하는 압축 알고리즘의 인터페이스

Fig. 11. Interface between external frame memory and proposed compression block

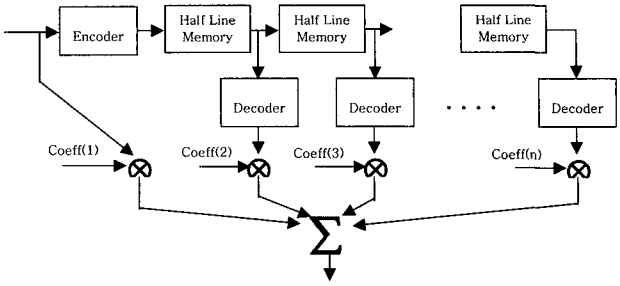


그림 12. 제안하는 압축 방법을 적용한 선형 필터의 구조

Fig. 12. Linear filter architecture adopting the proposed compression method

표 4. 선형필터에 적용한 경우의 메모리 절약 효과

Table 4. Memory saving effect for liner filter implementation

MN	360	720	1280	1920
1	-273 (1.25)	267 (0.88)	1,107 (0.71)	2,067 (0.64)
2	-72 (1.03)	1,008 (0.77)	2,688 (0.65)	11,520 (0.60)
3	129 (0.96)	1,749 (0.73)	4,269 (0.63)	7,149 (0.59)
4	330 (0.92)	2,490 (0.71)	5,850 (0.62)	9,690 (0.58)
5	531 (0.90)	3,231 (0.70)	7,431 (0.61)	12,231 (0.57)
6	732 (0.89)	3,972 (0.69)	9,012 (0.60)	14,772 (0.57)
7	933 (0.88)	4,713 (0.68)	10,593 (0.60)	17,313 (0.57)
.	.	.	.	.
32	5,958 (0.82)	23,238 (0.66)	50,118 (0.59)	80,838 (0.56)

VI. 결 론

본 논문에서는 멀티미디어 프로세서의 경쟁력을 높이기 위해 사용되는 메모리를 반으로 줄이며 영상의 화질을 유지하면서 하드웨어적으로 쉽게 구현이 가능한 영상 압축 알고리즘을 제안하였다. 래스터 스캔의 입출력을 지원하기 위해 1x8 블록 단위로 처리를 하였으며 압축의 효율성을 높이기 위해 하드웨어적으로 쉽게 구현할 수 있는 Hadamard 변환을 적용하였으며 변환 후에 계수들의 분포 특성을 이용하여 양자화 이전에 계수의 특성에 따라 thresholding을 적용하였다. 양자화는 각 계수들의 손실을 최소화 할 수 있는 양자화 레벨을 결정하였다. 또한 제안하는 알고리즘을 하드웨어로 구현하여 요구되는 로직의 양을 분석하고 실제 멀티미디어 프로세서에 적용했을 때 미치는 효과에 대해 체계적으로 분석하였다. 결론적으로 제안하는 알고

리즘은 우수한 성능을 보이며 이러한 알고리즘을 메모리를 필요로 하는 멀티미디어 프로세서에 적용한다면 칩의 경쟁력을 높일 수가 있을 것이다.

참 고 문 헌

[1] Rafael C. Gonzalez, *Digital Image Processing*, Prentice-Hall.

[2] N. Jayant and P. Noll, *Digital coding of Waveforms*, Prentice-Hall.

[3] Khalid Sayood, *Instruction to Data Compression*, Morgan Kaufmann Publishers.

[4] Seung-Jong Choi, Jin_Gyeong Kim, Hwa-Young Lyu, and Jong-Seok Park, "Frame Memory Reduction for MPEG-2/DTV Vodeo Decoding," *International Workshop on HDTV '98*, pp. 293-300, Oct. 1998

[5] Dong-Ho Lee, Jong-Seok Park and Yung-Gil Kim, "HDTV video decoder which can be implemented with low complexity," *IEEE International Conference on Consumer Electronics*,,pp. 6-7, 1994.

[6] N. Memon, "Adaptive coding of DCT coefficients by Golomb-Rice codes," *International Conference on Image Processing*, vol. 1,pp.516~520, 1998

[7] R. Bruni, A. Chimenti, M. Lucenteforte, D. Pau and R. Sannino, "A novel adaptive vector quantization method for memory reduction in MPEG-2 HDTV decoders," *IEEE International Conference on Consumer Electronics*, pp. 58-59, 1990

[8] Macro Winzker, Peter Pirsch and Jochen Reimers, "Architecture and memory requirements for stand-alone and hierarchical MPEG HDTV-decoders with synchronous DRAMs," *IEEE International Symposium on Circuits and Systems*, vol. 1, pp.609~612, 1995

[9] Anthony Vetro, Huifang Sun, Jay Bao and Tommy Poon, "Rrequency domain down conversion of HDTV using adaptive motion compensation" *International Conference on Image Processing*, vol.1, pp. 763~766,1997

[10] Wenwu Zhu, Kyeong Ho Yang and Farid A. Faryar, "A fast and memory efficient algorithm for down- conversion of an HDTV bitstream to an SDTV signal," *IEEE Transactions on Consumer Electronics*, vol. 45, no. 1, pp. 57~61, 1999.

저 자 소 개

정 수 운(학생회원)

1996년 2월 한양대학교 공학대학 제어계측공학과 학사, 1998년 2월 한양대학교 제어계측공학과 석사, 현재 한양대학교 전자전기제어공학과 박사과정 재학중. <주관심분야 : 멀티미디어 시스템, VLSI 설계, 버스 기반의 디지털 시스템 등.>

김 이 랑(학생회원)

2001년 2월 한양대학교 공학대학 전자컴퓨터공학부 학사, 현재 한양대학교 전자전기제어공학과 석사 과정, <주관심분야 : 영상 처리 및 압축, VLSI 설계, DTV 등.>

이 동 호(정회원)

대한전자공학회 논문집 제40권 CI편 제6호 참조