제22회 한국정보처리학회 추계학술발표대회 논문집 제11권 제2호 (2004. 11)

영상압축을 위한 SPIHT 알고리즘의 효율적인 하드웨어 설계

유 몽, 송문빈, 정연모
경희대학교 전자공학과

e-mail:evamong@hotmail.com

Efficient Hardware Design of SPIHT Algorithm for Image Compression

Mong Yu, Moonbin Song, Yunmo Chung
Dept of Electronic Engineering, Kyung-Hee University

요 약

This paper proposes an efficient hardware implementation of SPIHT(Set Partitoning In Hierarchical Tree) algorithm for image compression with the discrete wavelet transform. An efficient technique to scan the coefficients which are located in partitioned spatial orientation trees by DWT is considered in terms of counter fields for sorting pass and refinement pass. The proposed image compression method using SPIHT has been modeled in VHDL and has been implemented by use of both TMS320C6000 as a DSP and Virtex2 as a Xilinx FPGA.

1. 서론

멀티미디어 시스템의 빠른 발전으로 네트워크 상에서 다양한 정지 또는 동영상을 제공하는 서비스는 급속히 늘고 있다. 그러나 수신단의 대역폭, 계산능력, 화면 출력시간 등이 다양하기 때문에 효율적으로 영상을 전송하기 위한 부호화 기법이 필요하다.

오늘날 정지영상 압축 분야에서 가장 많이 사용되는 JPEG(Joint Picture Expert Group)의 단점을 보완하기 위해 DCT 변환 대신 웨이블릿(wavelet) 변환을 사용하는 연구가 진행되고 있다. 웨이블릿 변환된 영상의 효율적인 부호화 방법이 많이 연구되어 다양한 알고리즘이 사용되고 있다. 실제로 정지 영상 압축 분야의 새로운 표준인 JPEG2000에서는 웨이블릿 변환을 사용하고 있다[1]. 이는 JPEG에서 가장 문제가 되어왔던 블록 단위의 DCT 변환 때문에 발생하는 블록화 현상을 방지할 수 있기 때문이다.

SPIHT부호화는 웨이블릿 변환된 계수값들 중에서 크기 순으로 정렬하는 과정을 통하여 이루어진다. 기본적으로 부호화 하려는 대상에서 어느 한 계수라도 임계치(threshold) 보다 값이 크면 '1'을 전송하고, 모든 계수가 임계치 보다 값이 작으면 '0'을 전송한다. 그리고 다시 임계치를 반으로 줄여 이 과정을 반복하면서 계수들을 크기순으로 정렬하여 정밀화하는 과정을 수행한다[4].

본 논문에서는 SPIHT를 구성하는 제로트리(zerotree)개념과 대역간의 유사성을 이용하여 효율적으로 영상을 계층형 부호화하고 이를 하드웨어로 구현하는 연구를 하였다.

2. 이론

2차원 영상을 웨이블릿 변환하기 위해서 필터뱅크 구조를 구성한다. 필터뱅크는 고주파필터와 저주파필터로 구성된다. 입력영상을 필터뱅크를 통과시키면 고주파 및 저주파필터 처리된 두개의 영상을 생성한다. 이들을 다시 필터뱅크 처리하면 [그림 1]과

같이 4 개의 결과 영상 HH, HL, LH, LL를 얻을 수 있다. 그림은 2단계 필터뱅크 처리된 결과이다.

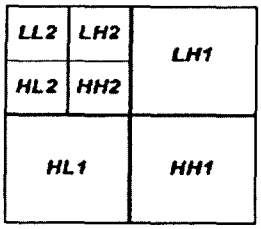

그림 1. 2 단계 분해

이러한 과정을 거쳐 저주파 대역만을 분해하는 것을 옥타브 분해법(octave decomposition)이라고 한다[4]. [그림 2]는 2단계 필터뱅크 처리를 거친 영상을 나타내었다. 좌측은 입력영상이고 우측은 2단계 필터뱅크 처리한 결과이다. 필터뱅크 과정을 웨이블릿변환 이라고 한다.

그림 2. 2-레벨 웨이블릿 분해된 영상

제로트리 부호화는 웨이블릿 변환된 영상에서 부대역간의 유사성을 이용하여 압축하는 방식이다. 웨이블릿변환 단계에서 상위 단계의 임계값이 임계값보다 작다면 하위 단계의 계수값도 작을 가능성이 높다는 사실을 이용하여 압축 효율을 높인다. 본 논문에서는 제로트리 부호화의 압축률을 높인 SPIHT를 기초로 한다.

SPIHT는 정렬 패스(sorting pass)와 세분 패스(refinement pass)의 두 단계의 연산으로 이루어진다. 정렬 패스에서는 계수들을 특정 임계값과 비교해서 중요도(significant or insignificant)를 알아내고 세분 패스는 정렬 패스에서 구한 중요한 계수값을 더욱 세밀하게 만든다[4].

변환된 계수값들을 스캔할 때 중요도를 판별하기 위해 비교를 수행할 입력 임계값은 (식 1)과 같이 정의할 수 있다. MAX(•)은 최대계수 값이며 γ(x,y)는 계수로 사용하고 최대 값보다 작은 2^n을 만족하는 가장 큰 값이다[6].

$$t_0 = 2^{\lfloor \log_2(MAX(|\gamma(x,y)|)) \rfloor} \quad (1)$$

SPIHT는 LSP(List of Significant Pixel), LIP(List of Insignificant Pixel), LIS(List of Insignificant Set)라는 세 개의 리스트를 사용해서 변환된 계수들을 관리한다. 초기에 LSP는 어떠한 계수값도 넣지 않으며 최상위 영역에 속한 계수들을 LIP에 넣는다. LIP의 계수들 중에서 자손들(descendants)을 LIS에 넣는다. 정렬 패스에서 LIS를 차례대로 조사해 가면서 특정 임계값보다 큰 값은 LSP에 넣고 특정 임계값 보다 작은 값은 LIP에 넣는다. 이 세 가지의 리스트는 중요한 계수 좌표 및 계수값을 관리하고 알고리즘의 패스를 만드는데 중요하다.

3. 설 계

[그림 3]은 DWT를 거쳐서 나온 계수값들을 SPIHT를 적용하여 압축한 후 산술부호화 하는 전체적인 블록도이다

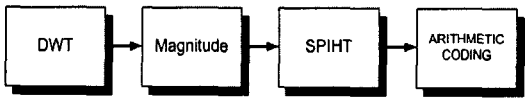

그림 3. 압축과정의 전체적인 블록도

[그림 4]는 SPIHT의 하드웨어 설계를 위한 블록도이다. 웨이블릿 변환을 거친 계수들이 주파수 대역별로 나누어져있는 상태에서 각 계수들을 스캔하고 이를 임계값과 비교하여 중요도를 검사하는 정렬 패스와 세분 패스를 거쳐서 LSP, LIP, LIS 에 구별하여 저장한다.

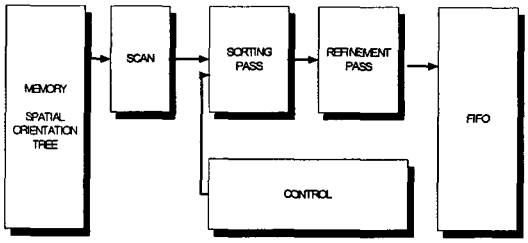

그림 4. SPIHT 전체 블럭도

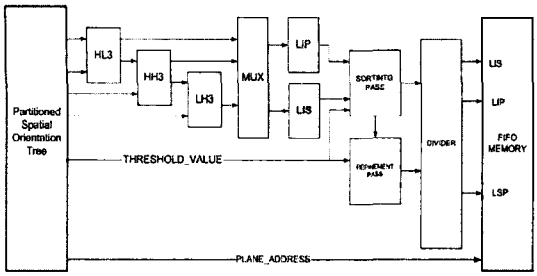

그림 5. SPIHT의 블록도

[그림 5]은 실제로 구현한 SPIHT의 하드웨어 구조이다. 계수값을 8x8 단위로 처리한다. 상위값들은 부모(parents)계수로 사용하고 그와 유사성을 가진 하위의 자손(descendants)계수들을 하나의 집합으로 간주하여 공간적 계층구조를 갖게 된다. 좌측의 HL3, HH3, LH3는 공간적 계층구조를 갖는 계수값들을 최상위에 있는 부모의 위치에 따라 분류하여 계층 집합별로 스캔하는 부분이다.

최상위에 부모 계수가 있으므로 이에 따라 3 개의 계층집합을 각각 스캔 한다. MUX블럭에서는 스캔 처리된 값들 중에서 최상위의 값들을 LIP에 넣은 후 그 자손들을 LIS에 넣는다. 그리고 임계값과 비교하여 중요도를 점검하는 정렬 패스와 세분 패스를 거친 값들을 Divider가 받아 처리한 후 메모리역할을 하는 다음블럭의 LIS, LIP, LSP에 분류하여 저장한다. 그 결과를 산술 부호화하여 최종 인코딩한다. SPIHT는 DCT 알고리즘과는 다르게 두 번의 패스단계를 거치므로 비록 부모 계수가 중요하지 않더라도 자손 계수 중에 중요한 계수가 있으면 이를 무시하지 않고 처리하여 부분적으로 하부영역의 값들을 부각시켜서 영상의 블록현상이나 왜곡현상을 줄인다.

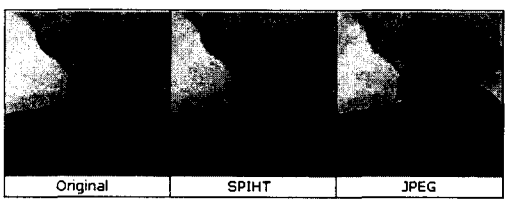

그림 6. 영상에서 SPIHT와 JPEG의 비교

[그림 6]는 원영상을 SPIHT 알고리즘을 이용하여 압축한 후 복원한 영상과 DCT 알고리즘을 이용하여 JPEG으로 압축한 후 복원한 영상을 비교한 그림이다. 그림에서와 같이 SPIHT를 적용한 영상이 JPEG영상보다 블록 현상이 적은 것을 알 수 있다.

4. 하드웨어 구현

본 논문에서 제안한 SPIHT를 이용한 영상압축 알고리즘을 [그림 7]과 같이 TI사의 TMS320C6000 DSP, Xilinx사의 Virtex2 FPGA를 사용하여 하드웨어 시스템으로 구현하였다. 여기서 DSP는 웨이블릿 변환을 수행하여 웨이블릿 계수 생성과 임계값을 결정하는 역할을 한다. FPGA는 생성된 웨이블릿 계수들을 8x8 픽셀단위로 스캔하여 중요도를 판단한 후 압축알고리즘을 수행하여 메모리에 저장하는 역할을 한다. 보드의 출력값은 오실로스코프(Oscilloscope)와 로직분석기(Logic Analizer)로 검증하였다.

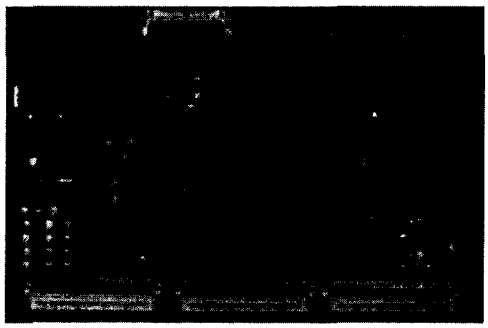

그림 7. SPIHT의 하드웨어 구현

5. 결론

SPIHT는 이산 웨이블릿 분해된 이미지의 효율적인 압축방법이다. 웨이블릿 변환의 특성인 자기 상관성을 바탕으로 정렬 패스와 세분 패스를 수행한다. SPIHT는 중요한 하부영역만 부호화하기 때문에 이 과정에서 발생하는 중복성을 제거하여 보다 효율적인 하드웨어 구조를 갖는다.

하드웨어 구현시 전체 픽셀값의 웨이블릿 변환을 적용하는 기능과 임계값을 설정하는 역할을 DSP가 담당하여 하드웨어적으로 처리하기 힘든 연산을 쉽게 소프트웨어적으로 처리하였으므로, FPGA에서는 웨이블릿 계수간의 중요도와 유사성을 판단하는 기능만을 수행하였다.

[표 1]은 SPIHT 알고리즘을 구현시 FPGA상에서 모든 처리를 하는 경우와 DSP와 FPGA를 병행적으로 사용하여 구현한 경우를 비교한 것이다.

표 1. SPIHT구현의 두가지 경우 비교

	FPGA상에서 모두 수행한 경우	DSP와 FPGA를 병행하여 수행한 경우
Chip Usage(%)	76.3	58.7
Logic Cell	2281	1742
Frame당 처리 clock수	142	69

DSP와 FPGA를 병행하여 사용한 경우가 그렇지 않은 경우보다 하드웨어 구현시 Chip usage가 작게 나오는 것을 알 수 있다. 그리고 Logic Cell의 사용량을 줄일 수 있고, Frame을 처리하기위한 clock의 숫자를 줄일 수 있다. 따라서 SPIHT알고리즘을 하드웨어로 보다 효율적으로 구현하였다.

참고문헌

[1] C. Christopoulos, A. Skodras and T. Ebrahimi, "The JPEG 2000 still image coding system:An overview," *IEEE Transactions on Consumer Electronics*, vol. 46, pp. 1103-1127, Nov. 2000.

[2] M. Rao et. al. *Wavelet transform Addison* Wesley, 1998.

[3] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on* Image Processing, vol. 1, pp. 205-220, April 1992.

[4] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on* Circuit and Systems for Video Technology, vol. 6, pp. 243-250, June 1996.

[5] A. Said, W. A. Pearlman, *SPIHT Image Compression: Properties of the Mathod.* Prentice hall , 1999.

[6] W. Knox Carey, et. al.
"Smoothness-constrained quantization for wavelet image compression," *IEEE Transactions on* Image *Processing,* vol. 8, No. 12, pp. 37-53 Dec. 1999.

[7] 이승훈, 윤동환, *Introduction to the Wavelet Transform*, 진한도서, 2002.

[8] T. Owen, S. Hauck, "Arithmetic Compression on SPIHT Encoded Image," University of Washington, Dept. of EE Technical Report UWEETR-2002-0007, 2002

[9] http://www.xilinx.com/

[10] http://www.ti.com/