

고속 수정 제로트리 알고리즘

준회원 김 호 식* · 이 복 훈*, 정회원 김 동 욱* · 유 지 상*

Fast Modified Zerotree Algorithm

Ho-sik Kim* · Bok-heun Lee* Associate Members, Dong-wook Kim · Ji-sang Yoo* Regular Members

요 약

본 논문에서는 Sharipo에 의해 소개된 EZW(embedded zerotree wavelet) 알고리즘을 개선하여 실시간 처리가 가능하며 효율이 더 좋은 방법을 제안하였다. 기존의 EZW는 각각의 심벌(symbol)마다 2비트를 할당하는데 이는 각 대역(band)에서 나타날 수 있는 심벌의 수를 고려하지 않고 부호화를 수행하는 것이므로 압축 효율이 떨어지는 단점이 있으며 부호화 과정에서 제로트리 코딩(zerotree coding)을 함으로써 많은 시간이 소요된다. 본 논문에서는 각 대역별로 심벌마다 가변 비트를 할당하여 압축 효율을 높였다. 그리고 Backscan방식을 사용하여 부호화 시간을 상당히 단축하였고 웨이블릿(wavelet) 변환을 리프팅 구조로 나타내어 전체적인 부호화 시간을 줄였다. 실험 결과 본 논문에서 제안한 방법이 기존의 EZW보다 PSNR은 약 0.3~1.5[dB], 부호화 시간은 2~10배 이상 향상됨을 확인하였다.

Key Words : wavelet; EZW.

ABSTRACT

In this paper, we propose a fast and efficient method that improves the performance of embedded zerotree wavelet(EZW) algorithm introduced by Sharipo. In the EZW algorithm, two bits are allocated for a symbol, but it is inefficient for compression and the zerotree coding wastes much time at encoding. In this paper, in order to increase the efficiency of compression, it will be allocated a variable bit for a symbol at each subband. To reduce the encoding time, we use a backscan method and lifting scheme instead of filter bank in wavelet transform. Experimental result are shown that the algorithm suggested in this paper has a better performance about 0.3~1.5dB PSNR while the encoding time was speeded up more than 2-10 times compared with the EZW algorithm.

I. 서 론

멀티미디어 정보는 저장이나 전송을 위해 대용량의 저장 매체나 큰 대역폭을 필요로 한다. 특히 멀티미디어 정보 중 영상은 압축을 하지 않을 경우 그 용량의 방대함으로 인해 저장이나 전송하는데 어려움이 있어 효율적인 압축 방법들이 많이 연구되어 왔다.

기존에는 DCT를 이용한 영상 압축 방법들이 많이 개발되어 왔고 실제로도 많이 쓰여 왔다. 그러나

DCT를 기반으로 한 영상 압축은 8×8크기의 블록 단위로 처리를 하므로 낮은 비트율에서 블록화 현상이 생기고 정확한 비트율 제어가 어려운 단점이 있었다. 이러한 문제점들을 해결하기 위해 많은 연구가 수행되었고 그 중 웨이블릿을 이용한 영상 압축이 많이 부각되었다[1][2]. 웨이블릿은 시간과 주파수 영역에서 동시에 두 가지 정보를 알 수 있는 장점이 있다. 또한 웨이블릿 변환은 영상을 주파수 영역에서 다해상도(multi-resolution) 분해와 대역분할 부호화(subband coding) 방법을 결합시킨 형태이

* 광운대학교 전자공학과 디지털 미디어 연구실(hskim@corelogic.co.kr, dogtags@image.gwu.ac.kr, dwkim@daisy.kw.ac.kr, jsyoo@daisy.kw.ac.kr), 논문번호 : 020093-0226, 접수일자 : 2002년 2월 26일

※본 연구는 한국과학재단 목적기초연구(R01-2001-000-00350-0)지원으로 수행되었습니다.

다. 이는 영상 압축을 위한 편리한 계층적 구조를 제공한다. 그리고 웨이블릿 변환은 영상을 블록 단위로 처리를 하지 않고 영상 전체에 대해 분해 작업을 하므로 블록화 현상이 생기지 않는다.

웨이블릿 변환을 이용한 부호화 방법 중에서 Sharipo가 제안한 제로트리 부호화(EZW)는 대역간의 상관관계를 이용하여 제로 트리(zero tree)를 구성하고 각 대역간의 자기 상관성에 기초한 연속적인 근사 양자화(successive approximation quantization : SAQ)를 이용한 방법이다[3]. 제로트리 부호화는 웨이블릿 계수와 임계값을 비교하여 중요 계수(significant coefficient)를 찾아내어 먼저 부호화를 시작한다. 이렇게 함으로써 임베디드 부호화가 가능하고 부호화 및 복호화에서 정확한 압축률 제어를 할 수 있다[3]-[5].

EZW 알고리즘은 웨이블릿 변환된 계수들을 4가지 심벌을 이용하여 부호화를 수행한다. 심벌은 중요 계수를 뜻하는 POS와 NEG가 있고 중요 계수가 아님을 뜻하는 IZ와 ZTR이 있다. EZW 알고리즘에서는 각각의 심벌마다 같은 길이의 비트를 할당한다. 그리고 EZW는 임의의 계수의 심벌을 나타내기 위해서 후손 계수들을 모두 읽어 들인다.

본 논문에서는 각 대역에서 존재하는 심벌의 수를 고려하여 심벌마다 가변 길이 비트를 적용하였고 웨이블릿 계수의 수직적인 관계뿐만이 아니라 수평적인 연관성까지 고려하여 제로블록 코딩(zero block coding)을 하였다. 그리고 임의의 계수를 부호화할 때 후손 계수들을 읽어 들이는 것이 아니라 부호화 전에 후손 계수들의 값을 미리 읽어 들여 버퍼에 저장하고 부호화에서는 단지 계수의 위치와 동일한 위치에 있는 버퍼에 저장된 값을 보고 심벌을 결정한다. 이는 상당한 시간적 단축을 가져온다. 또 웨이블릿 변환을 필터 뱅크(filter bank)방식이 아닌 리프팅(lifting)구조를 사용하여 빠른 변환을 가능하게 했다[6].

본 논문의 구성은 다음과 같다. II장에서는 EZW의 구체적인 알고리즘에 대하여 살펴보고 III장에서는 제안한 알고리즘을 설명하고, IV장에서 실험 및 결과를 제시하고, V장에서는 결론에 대해 언급한다.

II. Embedded Zerotree Wavelet

EZW는 웨이블릿으로 변환 후 다해상도를 갖는 영상의 압축에 많이 사용되어지는 효율적인 방법이다. EZW 알고리즘은 웨이블릿으로 분해된 영상을

부대역 별로 상위 대역을 부모(parent), 하위 대역을 자식(children)이라 정의하였고, '부모-자식(parent-children)'간의 자기 상관성에 기초한 연속적인 근사 양자화방법을 이용한 부호화 방법이다.

EZW는 임의의 비트율에서 부호화나 복호화를 멈출 수 있는 임베디드 특성을 갖고 있어 정확한 압축을 제어가 가능하다는 장점이 있다. 이러한 임베디드 특성을 갖으려면 현재 발생한 부호가 이후에 발생되어질 부호에 영향을 받지 않아야 된다. Sharipo는 제로트리라는 데이터 구조를 정의하여 계수들의 위치 정보를 효율적으로 나타냄으로써 항상 중요한 계수가 덜 중요한 계수들 보다 먼저 부호화가 되도록 하였다.

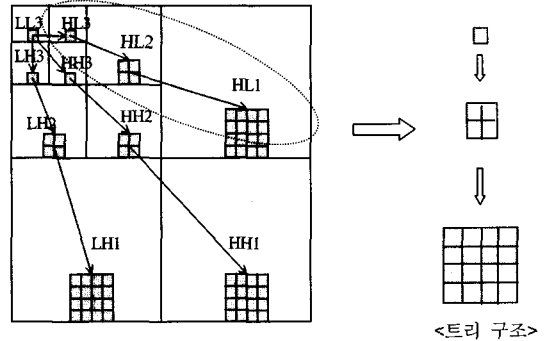


그림 1. 웨이블릿 변환된 영상에서의 트리 공간
Fig 1. Quadtree in wavelet transformed image space

그림 1은 3단계 웨이블릿 변환된 영상의 위치적 상관성에 대해 보여주고 있다. LL3의 하나의 계수는 나머지 대역 LH3, HL3, HH3의 같은 위치에 있는 계수들과 동일한 위치정보를 갖는다. 여기서 LL3는 부모가 되고 LH3, HL3, HH3는 자식이 된다. 그리고 LH3에 속한 하나의 계수는 LH2의 4개의 계수들과 동일한 위치 정보를 갖고 앞에서와 마찬가지로 LH3가 부모, LH2가 자식이 된다. 제로트리 부호화는 이와 같은 '부모-자식'의 관계에서 일반적으로 부모의 계수 값이 클 경우, 자식의 계수 값들도 클 확률이 높으며, 부모의 계수 값이 작으면, 자식의 계수 값들도 작을 확률이 높다는 성질을 이용한다. 즉, 현재 부호화하고자 하는 계수의 절대값이 임계값보다 크지를 판단하여 임계값보다 크다면 자식의 계수들까지 부호화를 하고 그렇지 않으면 자식의 계수값을 판단하여 부호화를 한다.

EZW 부호화는 크게 주 부호화 과정(dominant

pass)와 종속 부호화 과정(subordinate pass)을 거친다. 주 부호화는 임계값을 기준으로 중요 계수(significant coefficient)를 가려내는 과정이고, 종속 부호화는 중요 계수들의 정밀도를 높여주는 과정이다.

주 부호화 과정에서는 계수들의 상태에 따라 표 1과 같이 네 가지의 심벌들을 사용하여 부호화하고 이 때 중요한 계수로 판명이 되면 subordinate list에 그 값을 저장한다. 스캐닝은 그림 2와 같이 부모에서 자식방향으로 상위 대역에서 하위 대역 방향으로 진행된다.

표 1. 심벌의 종류
Table 1. A kind of symbol

심벌	심벌의 의미
POS (positive coefficient)	웨이블릿 계수의 절대값이 임계값보다 크며 양수일 때
NEG (negative coefficient)	웨이블릿 계수의 절대값이 임계값보다 크며 음수일 때
IZ (isolated zerotree)	웨이블릿 계수의 절대값이 임계값보다 작으나 자식 계수들중에 임계값보다 큰 값이 있을 때
ZTR (zerotree root)	웨이블릿 계수의 절대값이 임계값보다 작으며 자식 계수들의 값도 임계값보다 작을 때

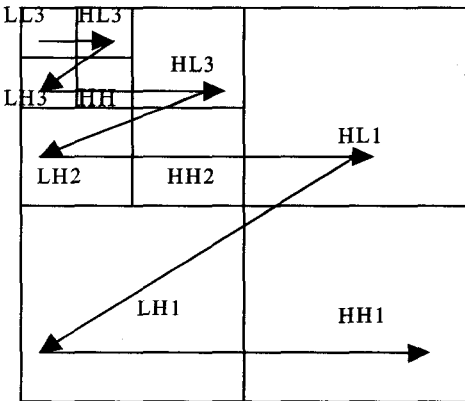


그림 6. 부호화를 위한 스캐닝 순서
Fig 2. Scanning order for the encoding

종속 부호화 과정에서는 주 부호화 과정에서 중요한 계수로 판명된 계수들에 대해 현재 임계값의 1/2보다 크면 '1'로, 작으면 '0'의 precision bit를 발생하여 원래 계수 값에 좀 더 근사화 시키는 연

속적인 근사 양자화를 수행한다. EZW 알고리즘을 수행하기 위해선 초기 임계값을 설정해야 하는데 초기 임계값은 전체 웨이블릿 계수들 중 최대값의 MSB(most significant bit)의 크기로 설정한다. 그리고 한 단계가 끝나면 임계값은 1/2배를 하여 같은 과정을 반복한다.

앞에서 말한 알고리즘을 간단히 요약하면 다음과 같다.

- (1) 초기 임계값(Threshold) 설정 :

$$T = 2^n, n = \lceil \log_2 [\max(|M(i, j)|)] \rceil$$

$\lceil x \rceil$: max{ n | n ≤ x, integer n }

$M(i, j)$: (i, j) 위치의 계수 값

- (2) 주 부호화 과정 :

심벌 출력 및 중요 계수 리스트 작성

- (3) 종속 부호화 과정 :

중요 계수 값들의 근사화

- (4) $T = T/2$, (2) 과정 반복

※ 과정 (2)에 대한 순서도가 그림 3에 나타나 있다.

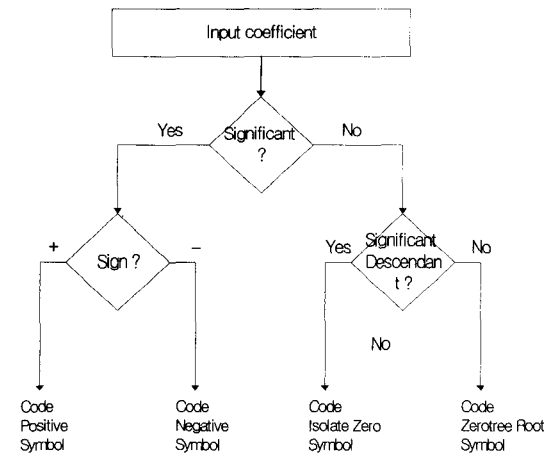


그림 3. 주 부호화 순서도
Fig 3. The dominant pass structure

EZW 알고리즘의 복호화는 먼저 초기 임계값을 읽어 들여 설정을 한다. 부호화와 마찬가지로 복호화도 주 부호화 과정과 종속 부호화 과정으로 나뉘고 각 과정에서 하는 동작은 부호화와 동일하다. 단지 부호화에서 심벌들이 발생하는 부분을 비트를 읽어 들이는 부분으로 바꾸어 주지만 하면 된다. 최종적으로 복원된 계수들은 양자화 오차를 줄여주기 위해 임계값의 절반을 더하거나 빼준다.

III. 고속 수정 제로트리 알고리즘(Fast Modified Zerotree Algorithm:FMEZW)

EZW 알고리즘을 이용하여 부호화할 경우는 대역에 상관없이 심벌마다 같은 비트를 할당하는 반면, 제안한 방법은 각 대역을 구분하여 심벌마다 가변 비트를 할당해 효율을 높였다. 그리고 제로 트리를 개선한 spatial orientation tree 구조를 사용하였다. 또, 제로 트리 부호화과정 중 부모를 부호화하기 위해선 자식 모두를 읽어 판단을 해야 되기 때문에 같은 계수 값들을 여러 번 읽어들이므로 실시간 처리가 어렵다. 제안한 방법은 backscan 방식을 사용함으로써 제로 트리 부호화에 걸리는 시간을 단축하였다.

3.1 가변 비트 할당(Variable Bit Allocation)

가변 비트 할당은 주 부호화 과정에서 발생하는 심벌의 비트를 대역에 따라 가변적으로 적용한 알고리즘이다. 그림 4와 같이 웨이블릿 변환된 영상의 대역을 4개의 구역으로 나눈다. 각 구역을 나누는 조건은 다음과 같다.

- ① 구역 : LL이면서, 루트가 아닌 경우,
- ② 구역 : LL이면서, 루트인 경우
- ③ 구역 : 루트가 아니면서, 자식을 갖는 경우 (LH3, HL3, HH3, LH2, HL2, HH2)
- ④ 구역 : 최하위 대역일 경우 (LH1, HL1, HH1)

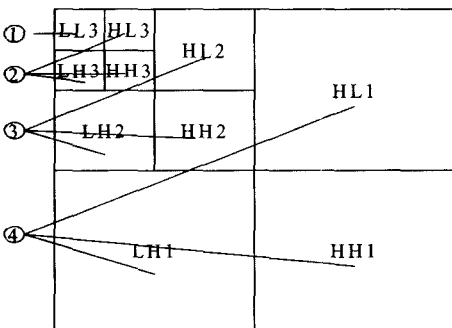


그림 4. 가변 길이 코딩을 위한 대역 설정
Fig 4. Assigning band for variable length coding

각 구역별로 부호화 방법을 알아보면, 먼저 LL이면서 루트가 아닌 ①번 구역은 발생할 수 있는 심

벌이 POS와 ZTR 두 개 뿐이다. 그러므로 이 구역은 각 심벌마다 1비트씩만을 할당하여도 복원 시에 아무런 문제가 없다. 그리고 이미 중요 계수로 판단이 되어 부호화가 된 계수는 부호화를 수행하지 않고 건너뛴다. 두 번째로, LL이면서 루트인 경우인 ②번 구역에서 발생할 수 있는 심벌은 POS, IZ와 ZTR 세 가지이다. 이 심벌들을 효율적으로 부호화하기 위해서 POS는 '10', IZ는 '11'로 각각 2비트를 할당하고 ZTR에 대해서는 '0'으로 1비트만을 할당하여 부호화를 수행한다. 그리고 이미 중요 계수로 판단이 되었으면 IZ나 ZTR로 부호화를 하는데 이 때도 발생하는 심벌이 2개 뿐 이므로 IZ에는 '1', ZTR은 '0'으로 1비트씩만을 할당한다. 세 번째로, 루트가 아니면서 자식을 갖는 경우인 ③번 구역은 EZW와 같은 방식으로 부호화를 수행한다. 각각의 심벌에 대해 2비트씩을 할당하고 이미 부호화가 된 계수에 대해서는 앞에서와 마찬가지로 발생하는 심벌이 IZ와 ZTR 두 가지 뿐이므로 각각 1비트씩만을 할당하여 부호화를 한다. 마지막으로, 최하위 대역인 ④번 구역은 자식을 갖고 있지 않으므로 IZ가 발생할 수 없다. 그러므로 발생할 수 있는 심벌은 POS와 NEG 그리고 ZTR 세 가지이다. 이 경우에도 효율적으로 부호화하기 위해 POS와 NEG에는 2비트를 할당하고 ZTR에는 1비트만을 할당하여 부호화를 수행한다.

3.2 제로 블록 코딩(Zero Block Coding)

EZW 알고리즘은 부호화를 할 때 주변 계수들과의 상관성은 고려하지 않고 수직적으로 하위 대역과의 관계만 고려하여 부호화를 한다. 그러나 일반적으로 하위 대역에서는 주변 계수들간에 연관성이 존재한다. 즉 임의의 계수 주변에서는 같은 심벌들이 많이 나타난다. 제로 블록 코딩(zero block coding)은 이런 수평적인 연관성을 이용하여 좀 더 효율적으로 부호화를 한다[7].

제안한 방법에서는 최하위 대역만을 고려하고 ZTR 심벌에 대해서만 처리를 하였다. 이는 최하위 대역에서 ZTR이 존재하는 경우가 많기 때문이다. 먼저, 최하위 대역을 2x2 블록들로 분할하고 각 블록들에 속한 계수들이 임계값보다 큰 값인지 작은 값인지를 조사한다. 만약 모두 임계값보다 작은 값이라면 그 블록 안에는 모두 ZTR만 있으므로 제로 블록으로 처리한다. 부호화 과정에서는 현재 블록이 제로 블록인지 아닌지를 알기 위해서 플래그 비트를 1비트 할당한다. 부호화 과정에선 이 플래그 비

트를 보고 이 블록이 어떤 블록인지를 구별할 수 있다.

최하위 대역에서 한 블록의 계수들이 모두 ZTR 일 때, EZW는 각각의 계수들을 따로 부호화하므로 총 8비트를 발생시키고, 제로 블록 코딩은 단지 플래그 비트만을 발생시킨다.

3.3 Backscan 방법

제로 트리 부호화는 각 계수를 부호화할 때 지식 계수들을 읽어 부호화를 한다. 이는 실시간 처리에 있어 많은 시간을 요하므로 미리 각 계수의 최대값을 읽어 들여 버퍼에 저장하는 방식을 사용하여 실시간 처리에 용이하게 했다. 버퍼에 필요한 메모리는 영상의 크기를 수평 방향으로 X, 수직 방향으로 Y라 했을 때 $(X \times Y)/4$ 가 된다. 사용되어지는 버퍼를 d_{max} 라 했을 경우 다음과 같은 식에 의해서 값이 구해진다[8].

$$d_{max}[i][j] = \max \{ C[2*i][2*j], C[2*i][2*j+1], C[2*i+1][2*j], C[2*i+1][2*j+1] \}$$

$$i, j \geq X/4, Y/4 \quad \dots (1)$$

$$d_{max}[i][j] = \max \{ d_{max}[2*i][2*j], d_{max}[2*i][2*j+1], d_{max}[2*i+1][2*j], d_{max}[2*i+1][2*j+1], C[2*i][2*j], C[2*i][2*j+1], C[2*i+1][2*j], C[2*i+1][2*j+1] \} \quad i, j < X/4, Y/4 \quad \dots (2)$$

위 식에서 $C[i][j]$ 는 (i, j) 위치의 웨이블릿 계수를 나타낸다. d_{max} 는 하위 대역에서 상위 대역 쪽으로 구한다. 이렇게 역방향 순서로 읽어 들이면 각 계수들을 한 번씩만 읽어 들이게 된다. 식 (1)을 이용해 최하위 대역의 웨이블릿 계수 중 최대값을 d_{max} 로 설정하고 그 위의 대역은 식 (2)를 이용하여 d_{max} 를 구한다. 이 식에서와 같이 d_{max} 값을 미리 구해 놓으면 제로 트리 부호화 과정에서 지식

계수들 값을 판별할 때 단지 임계값과 현재 위치의 d_{max} 값만을 비교하면 된다.

전체적인 제안 알고리즘의 주 부호화 순서도가 그림 5에 나타나있다.

제안 알고리즘의 종속 부호화는 EZW와 동일한 종속 부호기를 사용한다.

부호화 과정은 EZW와 마찬가지로 먼저 초기 임계값을 받아들인다. 그리고 동일한 주 부호화 과정과 종속 부호화 과정을 따르고 부호화에서 비트를 발생시키는 부분을 입력받는 부분으로 대체시켜준다. 그런데 복호화 과정은 제로트리 과정이 필요치 않기 때문에 제안 알고리즘에 사용한 지식 계수들의 최대값을 미리 읽어 들여 들여 저장하는 backscan 과정이 생략된다. 복호화가 다 끝난 후에는 복원된 계수의 오차를 줄여주기 위해 마지막 임계값의 절반을 계수가 양수면 더해주고 음수면 빼준다.

IV. 실험 및 결과

본 논문에서는 512×512 크기의 Lenna, Barbara, Goldhill 등의 영상을 가지고 테스트를 하였고, 웨이블릿 변환은 영상 압축에 효율이 좋은 Anotonini(9,7) 필터를 리프팅 구조로 나타내어 사용하였고 다섯 단계 레벨로 변환을 하였다[9].

그림 6은 산술 부호화(arithmetic coding)를 하지 않은 EZW와 제안한 알고리즘인 FMEZW(Fast Modified EZW)의 PSNR을 나타낸 것이다. 그림에서 점선이 EZW의 결과이고 실선이 FMEZW의 결과이다. 전체적으로 FMEZW의 PSNR이 약 0.3~1.5[dB] 정도 기존의 EZW 보다 향상되었음을 알 수 있다. 그림 7은 본 실험에 사용된 영상들을 나타내었고 그림 8은 32:1의 압축비에 대해 EZW와 FMEZW 알고리즘을 적용하여 복원한 영상들을 보였다. Lenna 영상의 경우 32:1의 압축비에서 EZW 보다 제안 알고리즘의 성능이 향상됨을 시각적으로 확인할 수 있으나 Barbara와 Goldhill 영상들은 PSNR이 0.5dB정도 향상되었기 때문에 시각적으로 구분하기 어렵다. 표 2에는 EZW와 FMEZW의 부호화 시간을 나타내었다. Lenna 영상의 경우 임계값이 32가 될 때까지 부호화를 하면 PSNR이 30dB 정도가 나온다. 이 때 부호화 시간을 보면 EZW보다 10배 이상 단축되었음을 알 수 있다.

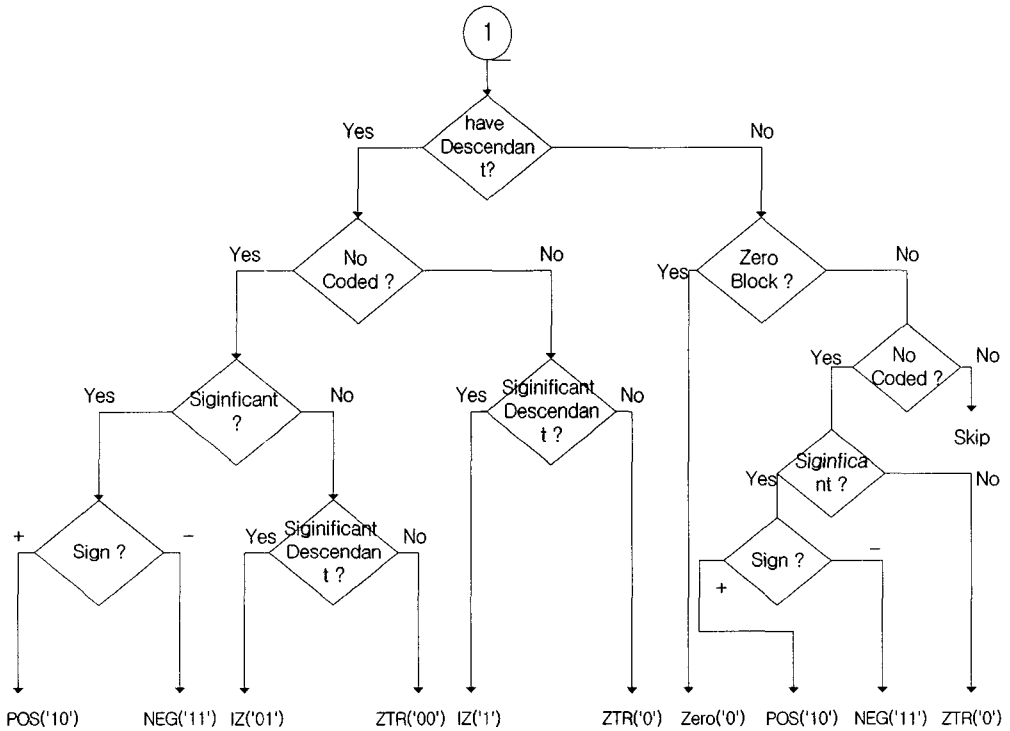
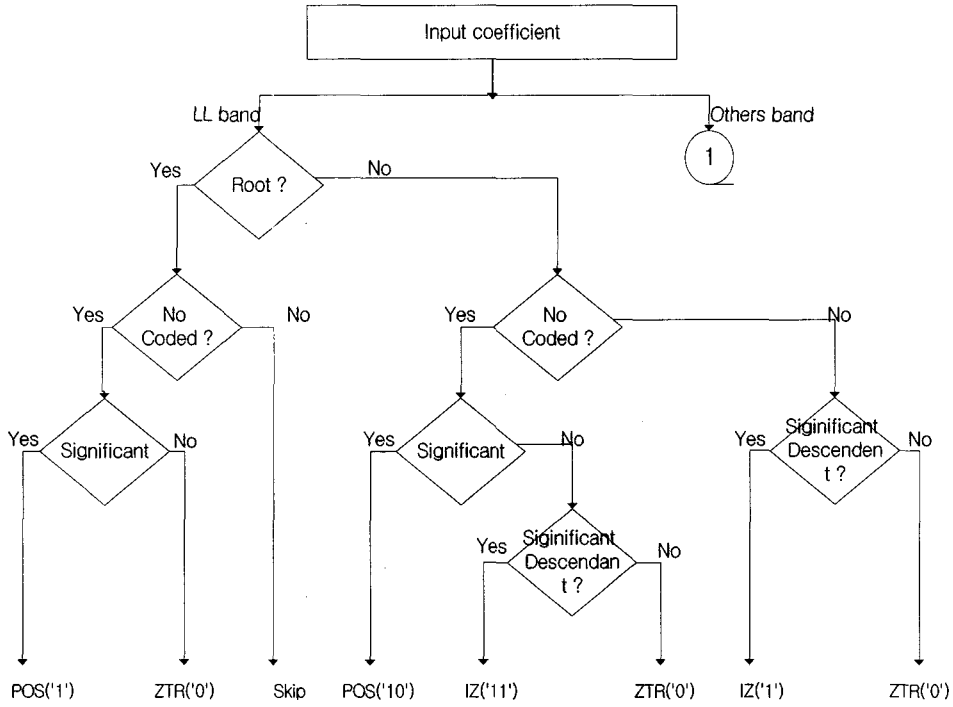


그림 5. 제안 알고리즘의 주 부호화 순서도
 Fig 5. The dominant pass of proposal algorithm

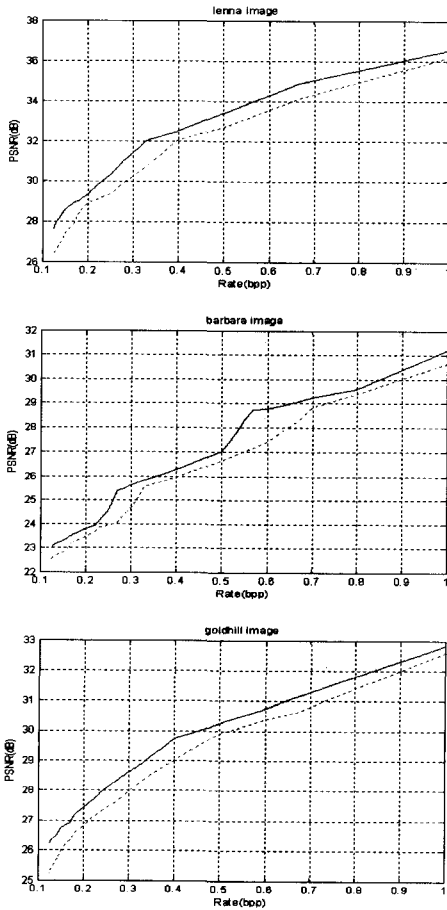


그림 6. 테스트 영상의 PSNR 비교
Fig 6. PSNR comparison for the Lenna image

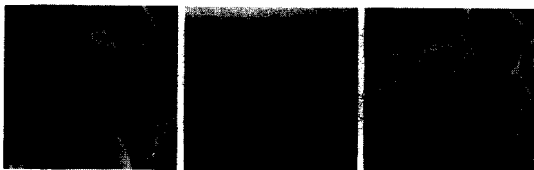
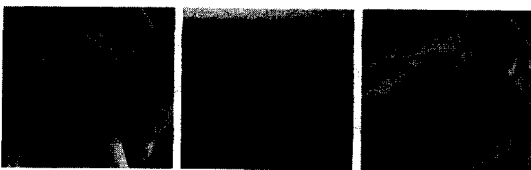
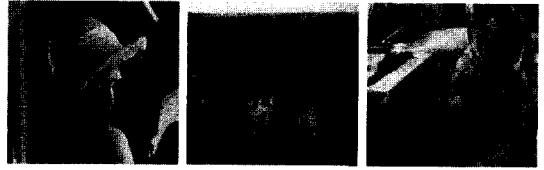


그림 7. 실험에 사용된 Lenna, Goldhill, Barbara 영상(좌측부터)
Fig 7. original test image



(a) EZW 알고리즘



(b) 제안된 FMEZW 알고리즘
그림 8. 32:1 압축비에서 EZW와 제안 알고리즘(FMEZW)의 복원 영상
Fig 8. The reconstructed image comparison with 32:1 compression ration

표 2. EZW와 FMEZW의 부호화 시간(sec)
Table 2. The Encoding time for EZW and FMEZW

	Lenna		Barbara		Goldhill	
	EZW	FMEZW	EZW	FMEZW	EZW	FMEZW
Threshold = 2	0.765	0.340	0.844	0.453	0.860	0.469
Threshold = 32	0.265	0.02	0.282	0.062	0.265	0.047

V. 결론

본 논문에서는 기존의 EZW 알고리즘을 분석한 후 대역별로 심벌마다 가변 비트를 할당하고 계수들간의 수평적인 관계까지 고려한 제로 블록 코딩을 적용함으로써 제안 알고리즘의 효율성을 높였다. 그리고 계수들의 값을 미리 읽어들이어 최대값을 버퍼에 저장하는 방식을 사용하여 제로 트리 부호화에 걸리는 시간을 상당히 단축하였다.

본 논문에서는 제로 블록 코딩의 경우 최하위 대역만을 고려하였는데 상위 대역까지 제로 블록 코딩을 적용하면 좀 더 향상된 결과를 얻을 수 있을 것으로 보인다. 그리고 제안 알고리즘은 각 심벌에 임의의 비트를 할당했기 때문에 부호기에서 발생한 비트열에 존재하는 0과 1의 수가 비슷하다. 이는 엔트로피(entropy) 코딩을 할 때 안 좋은 압축율을 보이므로 각 심벌에 적절한 비트를 할당하여 엔트로피 코딩의 압축율을 높일 수 있을 것이다.

참고 문헌

[1] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 205-220, Apr. 1992

[2] Amir Averbuch, Danny Lazar, and Moshe

Israeli, "Image Compression Using Wavelet Transform and Multiresolution Decomposition," *IEEE Trans. Image Processing*, vol.5, pp.4-15, Jan. 1996.

[3] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445-3463, Dec. 1993

[4] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol.6, pp.243-250, Jun. 1996.

[5] Zixiang Xiong, Kannan Ramchandran, and Michael T. Orchard, "Space-Frequency Quantization for Wavelet Image Coding," *IEEE Trans. Image Processing*, vol.6, pp.677-693, May. 1997.

[6] I. Daubechies and W. Swledens, "Factoring wavelet and subband transforms into lifting steps," *Journal of Fourier Analysis and Applications*, 4(3):245-267, Oct. 1998.

[7] 권진수, 유지상, "수정 블록 제로트리를 이용한 영상 압축," *한국통신학회 논문지*, 제26권 제4B호, pp. 471-478, 2001년 4월.

[8] F.W. Wheeler and W.A. Pearlman "SPIHT image compression without lists," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2047-2050, June 2000.

[9] J. D. Villasenor, B. Bellzer, and J. Liao, "Wavelet filter evaluation for image compression," *IEEE Trans. Image Processing*, vol. 4, no. 8, pp. 1053-1060, Aug. 1995.

김 호 식(Ho-sik Kim) 준회원



2001년 2월 : 광운대학교
제어 계측 공학과 졸업
2003년 2월 : 광운대학교 대학원
전자공학과 석사
2003년 3월 ~ 현재 :
CoreLogic software team 연구원
<관심분야> 웨이블릿 기반 영상
처리, 동영상 코덱, 영상통신

이 복 훈(Bok-heun Lee) 준회원



2003년 2월 : 광운대학교 전자
공학과 졸업(공학사)
2004년 3월 ~ 현재 :
광운대학교 전자공학과
석사과정

<관심분야> 웨이블릿 기반 영상처리, 동영상 코덱, 영상통신

김 동 욱(Dong-wook Kim) 정회원



1983년 2월 : 한양대학교
전자공학과 졸업(공학사).
1985년 2월 : 한양대학교
대학원 전자공학과 졸업
(공학석사)
1991년 9월 : Georgia 공과대학
전기공학과 졸업(공학박사)
1992년 3월~현재 : 광운대학교

전자재료공학과 정교수. 광운대학교 신기술 연구소 연구원

1997년 12월 ~ 현재 : 광운대학교 IDEC 운영위원
2000년 3월 ~ 현재 : 인티스닷컴(주) 연구원

<관심분야> 디지털 VLSI Testability, VLSI CAD, DSP 설계, Wireless Communication

유 지 상(Ji-sang Yoo) 정회원



1985년 2월 : 서울대학교
전자공학과 졸업(공학사)
1987년 2월 : 서울대학교 대학원
전자공학과 졸업(공학석사)
1993년 5월 : Purdue 대학교
전기공학과 졸업(Ph.D.)
1993년 9월 ~ 1994년 8월 :

현대전자산업(주) 산전연구소 선임연구원
1994년 9월 ~ 1997년 8월 : 한림대학교

전자공학과 조교수

1997년 9월 ~ 2001년 8월 : 광운대학교

전자공학과 조교수

2001년 9월 ~ 현재 : 광운대학교 전자공학과 부교수

<관심분야> 웨이블릿 기반 영상처리, 영상압축, 영상인식, 비선형 신호처리