

웨이블릿 기반 대역별 독립 공간 부호화에 의한 영상 압축

박 효 서[†] · 박 상 주^{††}

요 약

웨이블릿을 이용한 분해와 복원은 영상의 압축에 널리 사용되는 도구이다. 웨이블릿 분해된 자연 영상은 원 영상의 동일 부분을 표현하는 대역간의 계수 값 사이에 높은 상관 관계를 갖고 있고, 하나의 대역 안의 인접한 계수들 사이에도 높은 상관 관계를 갖고 있다. EZW (Embedded Zerotree Wavelet)나 SPIHT(Set Partitioning in Hierarchical Trees)과 같이 성능이 우수하고 널리 사용되는 웨이블릿 기반 영상 압축은 웨이블릿 계수의 대역간 상관 관계를 이용하고 있다. Zero Tree에 기초한 이와 같은 방법들은 계수값의 부모 후손 관계를 이용하기 때문에 대역별 부호화를 수행할 수 없으며, 다중 해상도 복원이 필요한 응용 분야에 효과적으로 적용되어지지 못한다. 본 논문에서는 하나의 대역 내에서 인접한 계수들 간의 상관성을 이용한 대역 독립적 부호화 기법을 제안한다. 압축 성능은 SPIHT과 대동하거나 약간 우수한 것으로 나타났고, 다중해상도 복원에 효과적으로 적용할 수 있음을 확인하였다.

Wavelet based Image Compression with Band Independent Space Coding

Hyo Seo Park[†] · Sang Ju Park^{††}

ABSTRACT

Analysis and synthesis using wavelet transform is a widely used method in image compression. Wavelet coefficients of a decomposed natural image have high correlation not only between interband coefficients representing the same area of original image but also between adjacent coefficients in single band. Widely used wavelet based image compression techniques such as EZW (Embedded Zerotree Wavelet) and SPIHT (Set Partitioning in Hierarchical Trees) utilize the interband correlation in terms of zero tree. Zero tree based methods are not very effective for applications that require multi-resolution reconstruction because they cannot perform band by band encoding or decoding. In this paper, a new wavelet based image compression method utilizing intraband coefficient correlation is proposed. It has been shown that the performance of this method is almost the same or slightly better than that of SPIHT depending on the nature of input images. This method is also very well suited for applications that need multi-resolution reconstruction.

키워드 : 웨이블릿(wavelet), 압축(compression), 공간(space)

1. 서 론

오늘날 정지영상 압축 분야에서 가장 많이 사용하는 JPEG (Joint Picture Expert Group)의 단점을 보완 하기 위해 DCT 변환 대신에 웨이블릿 변환을 사용하는 기법에 대한 연구가 진행되어 왔으며[5, 8, 9], 아울러 웨이블릿 변환된 영상의 효율적인 부호화 방법 또한 연구가 진행되어 왔다. 실제로 정지 영상 압축 분야의 새로운 표준인 JPEG 2000에서는 웨이블릿 변환이 채택되었다. 이는 JPEG에서 가장 문제가 되어왔던 8×8의 블록 단위의 DCT변환 때문에 일어나는 블록화 현상이 블록 단위가 아닌 영상 전체에 대한 분해 작업을 하는 웨이블릿 변환에서는 일어나지 않기 때문이며, 또한 EZW(Embedded

Zerotree Wavelet), SPIHT(Set Partitioning in Hierarchical Trees)과 같은 알고리즘을 적용하면 JPEG보다 훨씬 낮은 비트율에서 JPEG과 동일한 PSNR(Peak Signal to Noise Ratio)을 얻을 수 있어 압축율면에서도 우수하기 때문이다[1, 4].

또한 EZW, SPIHT은 비트 단위 처리를 하므로 부호화 및 복호화의 과정 중 원하는 비트율에서 언제라도 멈출 수 있다는 특징이 있다. 이러한 SPIHT의 부호화는 웨이블릿 변환된 계수의 값들 중에서 크기 순으로 정렬을 하는 과정을 통하여 이루어지는데, 기본적으로 부호화 하려는 집합에서 어느 한 원소라도 임계치 보다 값이 크면 1을 전송하고, 모든 계수가 임계치 보다 값이 작으면 0을 전송하며, 다시 임계치 값을 반으로 줄여나가면서 이 과정을 반복하게 된다. 이러한 과정을 통하여 계수들을 크기순으로 정렬하여 이렇게 정렬된 계수들을 정밀화하는 과정을 수행하게 된다.

이 때 SPIHT에서는 계수의 정렬과정을 효과적으로 부호화

* 이 논문은 2001년도 두뇌한국21 사업에 의하여 지원되었음.

† 준 회 원 : (주)인투스테크놀로지

†† 정 회 원 : 홍익대학교 전자전기공학부 교수

논문접수 : 2000년 11월 17일, 심사완료 : 2001년 3월 6일

하기 위하여 ZeroTree구조를 사용하는데, 이러한 ZeroTree구조는 웨이블릿 변환된 영상의 특징을 효과적으로 이용하도록 하는 기법으로 특히 압축을 하는데 있어서 효과적인 방법으로 사용된다[4].

이러한 ZeroTree구조를 사용하여 부호화 할 경우 영상의 점진적인 복원(저진 영상에서 정밀한 영상으로 복원)이 효과적이라는 장점을 가지게 되는데, 영상의 다중 해상도의 복원에는 별로 효과적이지 못하다.

또한 ZeroTree구조는 영상의 저주파수 대역부터 부호화해 나가기 때문에, 저주파수 대역(Low frequency band)에 에너지가 집중되어 있는 영상을 부호화 하는 데에는 효율적이지만 그렇지 않은 영상에서는 효율이 떨어진다[1, 4].

본 논문에서는 ZeroTree구조를 사용한 SPIHT과 같이 점진적인 복원이 가능하면서 동시에 다중 해상도(Multi-Resolution) 복원을 효과적으로 수행할 수 있는 부호화 방법을 제안하였다. 저주파수 에너지가 집중되어 있는 영상 뿐 아니라 고주파수 에너지가 집중되어 있는 영상 또한 효과적으로 부호화 할 수 있다.

본 논문의 구성은 2장에서 웨이블릿 변환된 영상의 부호화 기법에 대해 소개하고, 3장에서 제안한 방법에 대하여 설명하고, 4장에서는 제안한 방법을 모의 실험을 통하여 SPIHT과 비교 분석하였고, 마지막으로 5장에서 결론을 맺는다.

2 웨이블릿 변환된 영상

2.1 ZeroTree구조의 EZW와 SPIHT

일반적으로 자연 영상의 경우 인접 픽셀간 상호 연관성을 갖게 되는데, 이것을 화소단위에서 확대하여 웨이블릿 변환 계수의 영역으로 볼 경우에도 역시 상호 연관성을 갖고 있다[1].

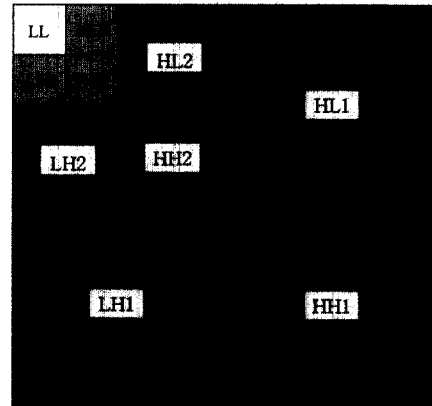
다시 말해서 현재의 영역이 강한 저주파수 성분을 갖고 있다면 그 인접한 영역 역시 강한 저주파수 성분을 갖고 있을 확률이 높고, 그 반대로 강한 고주파수 성분을 갖고 있다면 그 인접한 영역 역시 강한 고주파수 성분을 갖고 있을 확률이 높다는 것이다.

웨이블릿 변환된 영상을 부호화 하는 기본 원리는 일반적인 자연 영상이 저주파수 성분이 강한 특성을 갖고 있다는 사실을 이용한 것인데, 더불어 영상의 위와 같은 특징을 이용하면 보다 효과적인 부호화가 가능하다[2]. 웨이블릿 기반 영상 압축 기법의 대표적인 방법인 EZW와 SPIHT 역시 위와 같은 성질을 이용하고 있다[1, 4].

ZeroTree는 영상의 웨이블릿 변환에서 나타나는 부모-후손 관계에서 부모 계수의 절대값이 클 경우 후손 계수의 절대값 또한 클 확률이 높다는 사실을 이용한다. 부호화 순서 역시 영상의 저주파수 대역에서부터 시작하여 고주파수 대역으로 부호화해 나간다[1, 4]. 임계값을 설정하여 계수의 값이 임계값 보다 높은지 작은지를 부호화해 나가고, 계속

해서 임계값을 반으로 줄여가면서 반복하게 된다. 여기서 임계값 보다 높은 계수를 중요계수라 하고, 그렇지 않은 계수를 비중요계수라고 정의한다. ZeroTree는 이렇게 부모-후손 계수 관계를 이용하여 중요계수를 찾을 경우, LL대역을 중심으로 마치 Tree모양과 같이 뗫어 나가며, 어느 지점에서 이하 비중요계수라는 뜻의 Zerotree Root (End of Block)로 종료된다. 이러한 ZeroTree구조는 EZW에서 구제화 되었고, SPIHT에서는 ZeroTree 부호화 방법의 변화로 효율을 높였다. 자연 영상의 경우 저주파수 부분에 에너지가 집중되어 있는 형태를 취하기 때문에 이러한 ZeroTree구조를 이용하여 효율적인 부호화가 가능하다.

그러나 자연 영상에 이러한 특징이 있다 하더라도 항상 저주파수에만 에너지가 집중되어 있는 것은 아니기 때문에 고주파수에 에너지가 몰려 있는 영역을 부호화 하려 할 경우 효과적이지 못하다.



(a) 웨이블릿 변환된 영상의 대역



(b) Barbara영상을 3번 웨이블릿 분해
(그림 1) 웨이블릿 변환된 영상

예를 들어 (그림 1)의 (a)에서 HH2대역에 중요계수가 있지 않은 상태에서 HH1대역에 중요계수가 존재할 경우 ZeroTree의 구조상 HH1대역을 부호화 하기 위해선 HH2대역을 부호화 해야하는 단점을 가지고 있다. 이러한 비트의

낭비를 해결하기 위해서는 HH1대역의 HH2대역으로의 종속성을 제거해야 한다. 즉 HH2대역과 HH1대역을 따로 따로 부호화 해서 HH2대역을 부호화 하지 않으면서도 HH1대역의 부호화가 가능하도록 해야 한다.

따라서 대역간의 종속성을 제거하고, 각각의 대역을 하나의 독립된 공간으로 보고, 이 공간을 효과적으로 부호화 하는 공간부호화가 수행되어야 한다.

2.2 Zerotree구조의 다중해상도(Multi-Resolution)

영상 압축에서 Spatial scalability는 매우 중요한 문제 중의 하나이다. 이것은 하나의 부호화된 영상을 여러 서로 다른 장치에서 복호화하려고 할 경우 각각의 장치의 성능에 맞게 복호화가 가능하도록 하는 것으로서 영상의 크기가 다르게 나타날 수도 있고, 영상의 선명도가 다르게 나타날 수도 있어야 한다[6, 7].

ZeroTree를 사용한 부호화 방법의 경우 계수의 MSB (Most Significance Bit)부터 부호화 해 나가기 때문에 영상의 점진적인 부호화가 가능하게 된다. 하지만 ZeroTree의 부모 후손계수의 연관성 때문에 웨이블릿 변환된 계수의 하나의 비트플레인(bit plane)을 부호화 하는 과정에서 모든 대역이 뒤 쉼여 부호화가 진행되면서 부호화된 비트열(bit stream)의 경우 (그림 2)의 (a)와 같이 모든 대역이 뒤쉼여 나오게 된다[1, 4]. 따라서 ZeroTree를 사용한 기존의 방법(EZW, SPIHT)으로는 효과적인 다중해상도의 복원을 수행할 수 없게 된다.

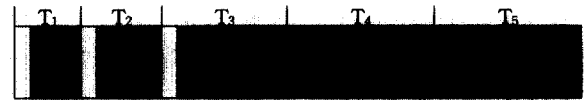
이것은 ZeroTree를 사용하지 않고, 각각의 대역을 독립적으로 부호화 함으로써 해결할 수 있다. 즉 대역간의 종속성을 이용하지 않고 각각의 대역을 독립하여 부호화 할 경우 (그림 2)의 (b)와 같이 LL대역부터 시작하여, HL3, LH3, HH3대역들을 같이 부호화하고, HL2, LH2, HH2대역들을 같이 부호화하고, 마지막으로 HL1, LH1, HH1대역들을 같이 부호화 함으로써 효과적인 다중해상도 복원이 가능하게 된다.

물론 기존의 방법 역시 복원하는 과정에서 영상의 크기를 자유로이 조절할 수 있고, 복원하려는 영상의 크기에 해당하는 메모리만 필요하지만, 이 경우 원 영상을 복원하는데 필요한 모든 처리 과정과 그 과정에서 사용되는 메모리가 필요하다. 예를 들어 SPIHT의 경우 LSP, LIP, LIS이라는 세 개의 목록을 사용하여 부호화 과정이 진행되는데, 부호화 과정동안 세 개의 목록에 계속해서 정보가 저장된다. 이것을 복원하기 위해선 영상의 크기에 상관없이 세 개의 목록에 저장되어 있는 모든 정보를 처리해야 하고, 그에 따라 원래의 크기를 갖는 영상을 복원하는데 사용되는 메모리가 필요하다.

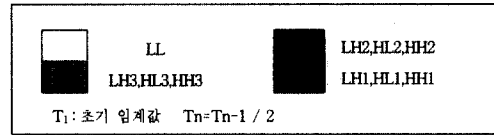
반면에 대역간의 연관성을 적용하지 않고, 각 대역을 독립하여 부호화 할 경우 영상의 복원에 필요한 대역만 선택하여 처리할 수 있으며, 복원에 필요한 메모리 역시 복원하려는 영상의 크기에 비례하여 필요하게 된다.



(a) ZeroTree를 사용한 비트열에서의 대역 분포



(b) 대역별 부호화를 수행하였을 때의 비트열



(그림 2) 비트열에서 대역 분포

2.3 다중해상도(Multi-Resolution)를 위한 대역별 부호화

웨이블릿 변환된 영상의 대역을 관찰해보면 각각의 대역에서 절대값이 큰 계수를 중심으로 에너지가 집중되어 있는 것을 알 수 있다. 즉 중요계수로 선택된 계수들을 중심으로 인접한 영역에서 계속해서 중요계수가 나올 확률이 높다는 뜻이다. 따라서 인접 영역간의 연관성을 적용하여 주어진 대역에서 값이 큰 계수를 몇 개 찾아낼 수만 있다면 그 영역을 중심으로 효과적인 부호화가 가능하다.

즉 부호화 과정에서 사용하는 임계값을 적용하여 몇 개의 중요계수를 찾은 후 이 중요계수들을 중심으로 인접 영역을 우선적으로 적절히 부호화 하게 되면 EZW와 SPIHT의 ZeroTree구조의 장점을 그대로 살릴 수가 있게 된다.

이렇게 대역간 연관성을 고려하지 않고, 하나의 대역 즉 하나의 공간 자체를 독립적으로 부호화 할 경우 고주파수 영역을 부호화 하는데 효과적이고, 대역별 부호화가 가능하기 때문에 영상의 다중 해상도 복원에 용이하다는 장점이 생긴다.

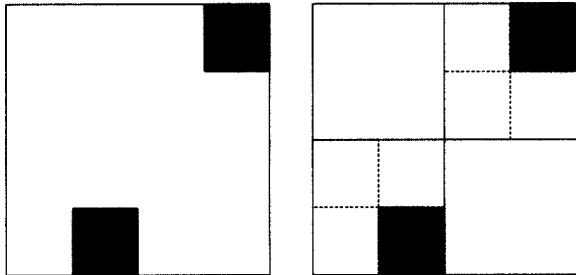
3 제안한 방법

3.1 공간부호화

웨이블릿 변환된 영상의 경우 계수의 절대값이 큰 점들을 중심으로 인접 계수들의 부호화가 진행되는 특징이 있기 때문에 부호화 과정 초기에서 중요계수로 선택된 계수가 있는 영역의 경우 인접한 영역들을 세밀하게 분해하여 부호화 함으로서 계속되는 부호화 과정에서 중요계수를 효과적으로 찾을 수 있게 하고, 그렇지 않은 영역들은 크게 나누어 부호화 함으로써 계속되는 부호화 과정에서 중요계수가 발생하지 않는 영역에 적은 비트가 할당되므로 효과적으로 부호화 할 수 있다.

본 논문에서는 웨이블릿 변환된 영상 각각의 대역을 독립된 공간으로 설정하고, 이 공간을 계속 분해해 나가면서 중요계수를 찾는 부호화 방법을 이하 '공간부호화'라 하였다.

이 공간부호화 방법의 구체적인 부호화 방법은 부호화 하려는 영역에 중요계수가 있으면 '1', 중요계수가 없으면 '0'으로 부호화하고, 중요계수가 있는 영역은 4개의 부분 영역으로 나누어 가며 부호화 하는 것이다.



(a) 중요계수의 위치 (b) 공간부호화
(그림 3) 공간부호화의 예

(그림 3)의 (a)에 두 개의 중요계수를 갖는 크기가 4×4 인 영역의 예를 보였다. 이것을 공간 부호화 방법을 사용하여 부호화 할 경우 (그림 3)의 (b)와 같이 영역이 분할된다.

먼저 영역 안에 중요계수가 있으므로 '1'이 출력되고, 현재 대역에 중요계수가 있으므로 전체 영역을 실선으로 보인바와 같이 작은 영역 4개로 나눈 후, 각각의 영역을 부호화 하면서 '0110'이 출력된다. 중요계수가 있는 영역을 다시 점선을 사용하여 더욱 작은 영역 4개로 나눈 후, 각각의 영역을 부호화 하면서 '0100'과 '0001'이 출력된다. 따라서 이 영역에서 중요계수 두 개를 부호화 하는데 발생한 비트열은 '1011001000001'이 된다.

```

부호화 하려는 대역의 크기가 2n×2n이라고 가정하자.
Step0 : make L + 2 empty lists : SL, SL-1, ..., S0 and SSIG
        ( L > 0 )
        put Current band into SL
        set n = [log2(max(coefficients of current band))]
Step1 : set l = L
        set th = 2l
Step2 : for each entry b in Sl
        If it contains significant coefficient(s) with respect to th
            output 1
            If l = 0
                move b into SSIG
            else
                divide b into 4 subbands b1, b2, b3, b4 and
                append those to Sl-1 remove b from Sl
            endif
        else
            output 0
        endif
Step3 : if l = 0
        goto Step4
    else
        set l = l - 1
        goto Step2
    endif
Step4 : send refinement bits for each entry except those
        included in the last Step2 b in SSIG
Step5 : set n = n - 1
        goto Step1
    
```

이러한 방법은 이미 언급한 웨이블릿 변환된 영상의 특성을 효과적으로 이용할 수 있다. 웨이블릿 분해된 자연 영상의 경우 중요계수 영역을 중심으로 계속해서 부호화가 일어날 확률이 높기 때문에 (그림 3)의 (b)에서와 같이 중요계수가 있는 영역을 중심으로 한 인접 영역을 세밀하게 분해함으로써 계속되는 부호화 과정에서도 효과적인 부호화가 이루어 질 수 있게 된다.

이러한 부호화의 방법을 구체적으로 정리해 보면 위와 같다.

3.2 다중 해상(Multi-Resolution)도 복원

웨이블릿 변환된 영상의 경우 서로 다른 여러 개의 대역이 생기게 되는데, EZW, SPIHT 등과 같이 ZeroTree를 사용하는 부호화 방법에서는 대역간의 연관성을 적용하여 부모-후손이라는 관계를 이용하였다. 즉 (그림 4)의 (a)와 같이 저주파수 대역에서부터 고주파수 대역에 이르기까지 영상의 같은 위치에 해당하는 부분을 연관지어 생각하였다[1, 4].

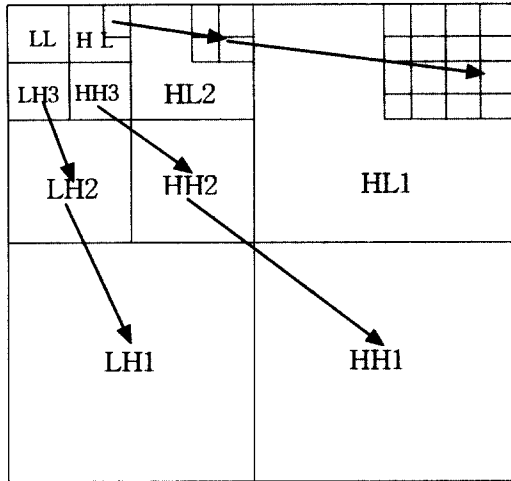
그러나 공간부호화 방법을 사용할 경우 이러한 대역간의 연관성을 고려하지 않고, (그림 4) (b)와 같이 각각의 대역을 독립된 영역으로 생각하여 처리할 수 있다.

공간 부호화 방법을 사용해서 부호화 할 경우 초기 임계값을 설정한 후 LL대역을 Step2에서 Step4까지 수행하고, 마찬가지로 HL3, LH3, HH3대역을 Step2에서 Step4까지 수행하고, 다음으로 HL2, LH2, HH2를 수행하고, HL1, LH1, HH1을 수행한 후, 임계값을 반으로 낮추어서 다시 반복한다.

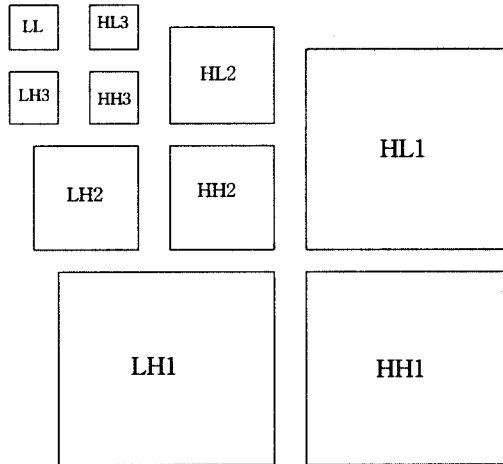
이 경우 해상도를 구분할 수 있는, 즉 HL3, LH3, HH3가 부호화 된 후 HL2, LH2, HH2가 부호화 되는 것을 구분할 수 있는 방법이 필요하게 되는데, 이것은 부호화 측에서 각각의 해상도에 해당하는 부호화 결과를 항상 버퍼에 저장해 두고 이것을 전송하기 전에 현재 부호화 된 비트열의 길이를 먼저 전송함으로써 해결할 수 있다.

예를 들어 현재 임계값이 주어지고, 그에 따른 중요계수를 찾는 과정을 살펴보면 (그림 4)의 (b)에서 LL대역을 공간 부호화 한다. 그런 후 HL3, LH3, HH3대역을 공간 부호화 한 후 이 과정에서 생성된 비트열의 길이를 먼저 전송한 후 결과를 전송한다. 마찬가지로 HL2, LH2, HH2대역을 공간 부호화 한 후 이 과정에서 생성된 비트열의 길이를 먼저 전송한 후 결과를 전송하고, HL1, LH1, HH1대역을 공간 부호화 한 후 이 과정에서 생성된 비트열의 길이를 먼저 전송한 후 결과를 전송하고, 마지막으로 임계값을 낮춘 후 LL대역부터 다시 공간 부호화 과정을 수행하면서 반복한다.

이 과정의 결과 생성된 비트열은 (그림 2) (b)와 같이 나오게 된다.



(a) 영상의 웨이블릿 변환시 대역간 연관성



(b) 공간부호화 적용시 각각의 대역을 하나의 독립된 공간으로 생각한다.

(그림 4) 영상의 웨이블릿 변환

3.3 점진적 복원(Progressive Transmission)을 위한 개선

2절의 방법은 다중해상도 복원을 위해서는 매우 효율적이지만 주어진 비트율에서 최상 화질의 영상을 복원하지는 못한다. 이것은 웨이블릿 계수 한 개를 부호화 할 때 발생한 비트만이 실제 하나의 계수값을 복원하는데 사용되기 때문이다. 즉 하나의 공간을 부호화 할 경우 중요계수의 위치를 전송하기 위해 공간을 계속 분해해 나아가야 하는데, 이 과정에서 발생하는 비트들은 최종 중요계수의 위치가 부호화 되기 전까지는 복원 영상의 화질 개선에 도움을 주지 못하기 때문이다.

결국 주어진 비트율에서 최적의 복원 영상을 얻기 위해서는 대역에 상관없이 크기가 항상 작은 공간부터 부호화 하고, 차음 큰 공간을 부호화 하는 것이 최적의 방법이 된다.

이러한 부호화의 방법을 구체적으로 정리해 보면 아래와 같다

부호화 하려는 영상의 크기가 $2^L \times 2^L$ 이라고 가정하자.

Step0 : make $L + 1$ empty lists : $S_{L-1}, S_{L-2}, \dots, S_0$ and S_{SIG}
($L > 0$)

put HL_x, LH_x and HH_x bands into S_{L-x}

put all coefficients in LL band to S_0

set $n = \lceil \log_2(\max\{\text{all coefficients in image}\}) \rceil$

set $k = 0$

Step1 : set $l = k$

set $th = 2^l$

Step2 : for each entry b in S_l

If it contains significant coefficient(s) with respect to th

output 1

If $l = 0$

move b into S_{SIG}

else

divide b into 4 subbands b_1, b_2, b_3, b_4 and

append those to S_{l-1}

remove b from the S_l

endif

else

output 0

endif

Step3 : if $l = 0$ then goto Step4

else

set $l = l - 1$

goto Step2

endif

Step4 : if $k = L-1$ then goto Step 5

else

set $k = k + 1$

goto Step1

end if

Step5 : send refinement bits for each entry except those included in the last Step2 b in S_{SIG}

Step6 : set $n = n - 1$

set $k = 0$

goto Step1

4. 모의실험 결과

4.1 모의실험

본 실험에서는 SPIHT과 제안한 방법을 실험하였고, 두 가지 방법을 비교하기 위해서 동일한 조건에서 실험을 하였다. 사용한 필터는 <표 1>에 보이는 9/7-tap 웨이블릿 변환 필터[5]를 사용하였고, 웨이블릿 변환 횟수는 6으로 하였다.

<표 1> 9/7 웨이블릿 변환 필터

n	0	±1	±2	±3	±4
$2^{-1/2}h_n$ (부호화)	0.602949	0.266864	-0.078223	-0.016864	0.026749
$2^{-1/2}h_n$ (복호화)	0.557543	0.295636	-0.028772	-0.045636	0

실험에 사용된 영상은 3가지로 512×512 Lena영상과 512×512 Barbara영상, 그리고 512×512 Goldhill영상을 사용하였고, 산술 부호화를 수행하지 않았다.

또한 SPIHT과 제안한 방법에서 발생하는 영상의 손실을 측정하기 위하여 PSNR을 사용하였다. 이때 PSNR의 식은 다음과 같다.

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) [dB]$$

먼저 다중 해상도 공간부호화(방법A)를 모의 실험하였고, 점진적 복원 공간부호화(방법B)를 모의 실험하였다.

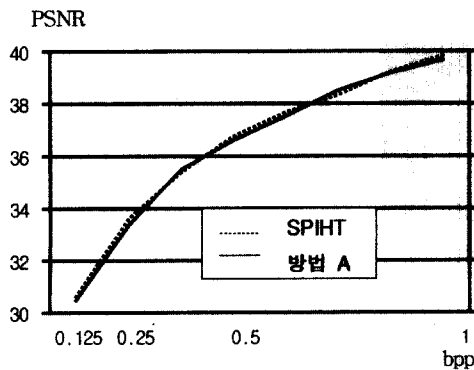
두 방법 모두 웨이블릿 변환에만 복잡한 연산이 수행되며, 변환된 영상을 부호화 할 때에는 계수들의 크기 비교만 수행하므로 별다른 연산이 필요 없기 때문에 수행 시간은 큰 차이가 없으며, 복원시 사용되는 버퍼도 알고리즘을 수행하는데 필요한 버퍼 외에 추가적인 버퍼가 사용되지 않으므로 얼마나 최적화된 프로그램을 만들어서 수행하느냐가 중요하다. 단 부호화시 두 방법 모두 속도를 높이기 위해 테이블을 위한 버퍼가 추가로 사용되는데, SPIHT의 경우 각 계수의 후손들 중 가장 큰 값을 적어 놓은 테이블이 필요하며, 제안한 방법A, B의 경우엔 각 공간에서 가장 큰 값을 적어 놓은 테이블 필요하다. 이것을 계산해 보면 $2^n \times 2^n$ 크기의 영상을 부호화 할 경우 SPIHT은 $(2^n \times 2^n)/4$ 크기의 테이블이 필요하며, 제안한 방법의 경우 약 $\approx \sum_{i=1}^n (2^i \times 2^i)/4$ 크기의 테이블이 필요한 차이가 있다.

4.2 다중 해상도 복원 공간부호화 방법과 SPIHT의 비교

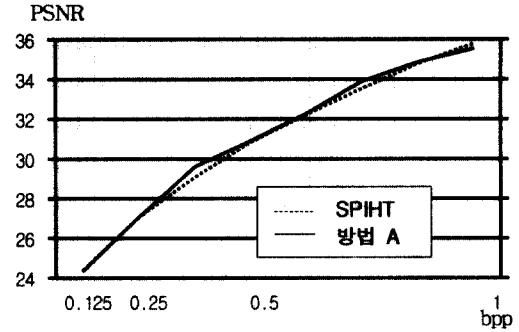
다중 해상도 복원을 효과적으로 수행하는 부호화 방법으로 부호화한 경우 각각의 해상도(Resolution)에 해당하는 비트열이 나오기 전에 비트(bit)의 개수가 먼저 부호화 되어야 한다.

이것으로 인해서 발생하는 오버헤드는 극히 작은 양으로 오버헤드를 제외한 결과와의 차이는 PSNR이 약 0.02정도 감소하는 정도로 무시할 수 있는 수치였다.

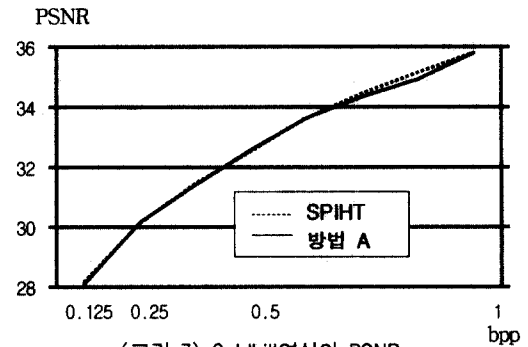
그 결과 (그림 5), (그림 6), (그림 7)에서 보는 바와 같이 Lena영상, Barbara영상, Goldhill영상 모두 SPIHT과 거의 비슷한 PSNR을 보이는 것을 알 수 있다.



(그림 5) Lena영상의 PSNR



(그림 6) Barbara영상의 PSNR



(그림 7) Goldhill영상의 PSNR

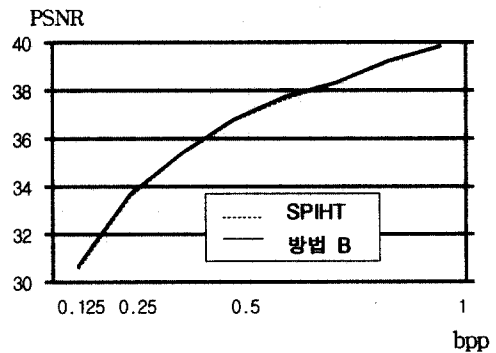
4.3 점진적 복원 공간 부호화 방법과 SPIHT의 비교

(그림 8)에서 보는 바와 같이 Lena영상의 경우 저주파수에 많은 에너지가 집중되어 있기 때문에 제안한 방법과 SPIHT의 PSNR이 거의 비슷하였다.

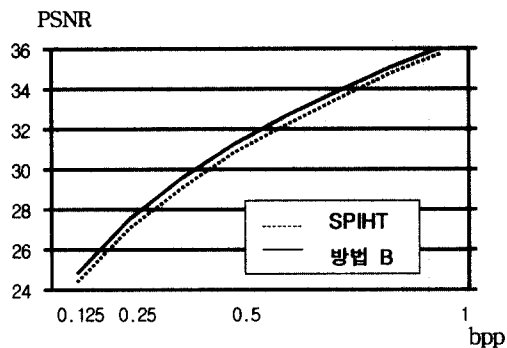
그러나 (그림 9)에서 보는 바와 같이 Barbara영상의 경우 고주파수에도 에너지가 집중되어 있기 때문에 제안한 방법의 PSNR이 SPIHT보다 약 0.3dB에서 0.5dB정도 우수한 성능을 나타냈다.

또한 (그림 10)에서 보는 바와 같이 Goldhill영상의 경우에 제안한 방법의 PSNR이 SPIHT보다 약 0.1dB정도 우수한 결과를 보였다.

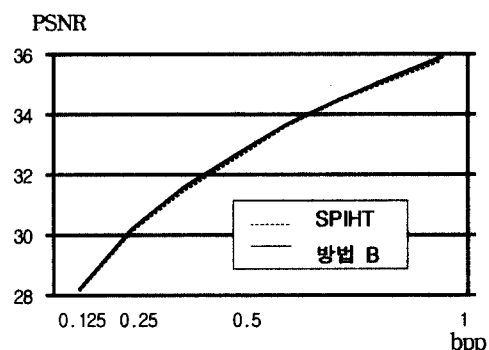
(그림 11)은 Barbara영상을 0.5bpp로 부호화 하였을 때 웨이블릿 분해된 영상이 부호화 과정 중 공간 부호화 된 결과를 나타내고 있다. 이 그림에서 Barbara영상의 경우 다리 부분에서 효율적인 공간 부호화가 이루어진 것을 알 수 있다.



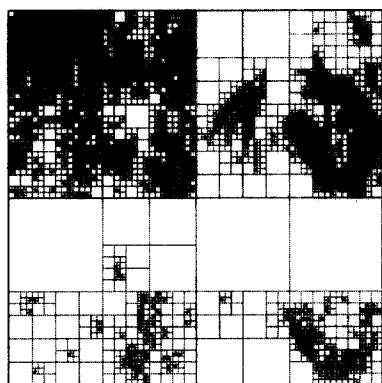
(그림 8) Lena영상의 PSNR



(그림 9) Barbara영상의 PSNR



(그림 10) Goldhill영상의 PSNR



(그림 11) Barbara영상을 0.5bpp로 부호화 하였을 때의 공간 부호화

5. 결 론

웨이블릿 변환된 영상을 효과적으로 부호화 하는 방법으로 EZW와 SPIHT등이 있으며, 이러한 부호화 방법의 경우 점진적인 복원이 가능하다. 그러나 다중 해상도 복원에 효과적이지 못하며, 고주파수 영역에 에너지가 몰려 있는 영상의 효과적인 부호화가 어렵다.

제안한 방법의 경우 기존 방법과 같이 점진적인 복원이 가능하면서도 PSNR의 손실 없이 다중 해상도 복원을 효과적으로 수행하였다. 특히 점진적인 복원에 효과적인 방법B의 경우 PSNR이 약 0.1dB정도 높게 측정된 것을 알 수 있었으며, 특정 영역에 에너지가 몰려 있는 경우 0.4dB이상

높게 측정되었다.

따라서 현재 사용하는 분야가 영상의 PSNR이 중요한 분야인 경우인지, 아니면 다중 해상도 복원이 더 필요한 경우인지를 판단하여 적절히 사용할 수 있는 방법을 제안하였다.

참 고 문 헌

- [1] J. M. Shapiro, "Embedded image coding using zerotrees of wavelets coefficients," *IEEE Trans. Signal Processing*, Vol.41, pp.3445-3462, Dec. 1993.
- [2] J. W. Woods and S. O'Neil, "Subband coding of images," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol.34, pp.1278-1288, October 1986.
- [3] G. Strang and T. Nguyen, "Wavelets and filter banks," Wellesley-Cambridge Press.
- [4] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits and Systems for Video Technology*, Vol.6, pp.243-250, June 1996.
- [5] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, Vol.1, pp.205-220, April 1992.
- [6] K. R. Rao and J. J. Hwang, *Techniques & Standards for Image Video & Audio Coding*, Prentice Hall, 1996.
- [7] K. H. Tzou, "Progressive image transmission : a review and comparison of techniques," *Optical Engineering* 1987, pp. 529-537.
- [8] Vetterli and Kovacevic, *Wavelets and Subband Coding*, Prentice Hall, 1995.
- [9] C. Sidney Burrus, *Introduction to Wavelets and Wavelet Transforms*, Prentice Hall, 1998.



박 호 서

e-mail : xylo@hitel.net

1999년 홍익대학교 전자전기 제어공학과 졸업(학사)

2001년 홍익대학교 대학원 전자공학과 (공학석사)

2001년~현재 (주)인투스테크놀로지, 연구원

관심분야 : 정지 영상 압축, 영상회의 시스템



박 상 주

e-mail : sjpark@wow.hongik.ac.kr

1984년 서울대학교 전자공학과(학사)

1986년 서울대학교 전자공학과(석사)

1992년 Stanford University, Electrical Eng., 박사

1992년~1993년 Philips Research Palo Alto, MTS

1993년~1995년 Philips Semiconductors, MTS

1996년~현재 홍익대학교, 전자전기공학부

관심분야 : 영상 처리 및 압축