오류에 강인한 제로트리 웨이블릿 영상 압축

정회원 장 우 영*, 송 환 종**, 손 광 훈**

An Error-Resilient Image Compression Based on the Zerotree Wavelet Algorithm

Woo Young Jang*, Hwang Jong Song**, Kwang Hoon Sohn** *Regular Members*

요 약

본 논문에서는 웨이블릿 변환을 이용한 오류에 강인한 영상 압축 기법을 제안하였다. 공간·주파수 영역에서 웨이블릿 계수들의 통계적 특성, 에너지 특성, 방향 특성을 이용한 제로트리 기법은 높은 압축 성능을 나타낸다. 하지만, 제로트리 부호화 기법은 부호에 따른 부호화 되는 계수의 수가 다르기 때문에 오류에 민감하게 반응하여 한 개의 오류가 전체 영상에 확산되어 영향을 미치게 된다. 제안 알고리듬에서는 SPIHT(Set Partitioning in Hierarchical Trees) 알고리듬을 이용한 제로트리 기법으로 영상을 부호화한다. 그리고 부호화 계수들을 부밴드간 상관도를 이용하여 비트열을 다수의 블록으로 분리하고 비트 재구성 알고리듬을 이용하여 같은 크기의 블록으로 만든다. 이 과정으로 효율적인 비트 할당과 오류의 전파를 해당 블록으로 제한하여 오류가 없거나 적은 환경에서는 제로트리 압축 기법과 유사한 성능을 보이며 오류가 많은 환경에서는 제로트리 압축 기법 및 기존의 오류에 강인한 압축보다 더 효율적으로 부호화 할 수 있다.

ABSTRACT

In this paper, an error-resilient image compression technique using wavelet transform is proposed. The zerotree technique that uses properties of statistics, energy and directions of wavelet coefficients in the space-frequency domain shows effective compression results. Since it is highly sensitive to the propagation of channel errors, even a single bit error degrades the whole image quality severely. In the proposed algorithm, the image is encoded by the SPIHT(Set Partitioning in Hierarchical Trees) algorithm using the zerotree coding technique. Encoded bitstreams are partitioned into some blocks using the subband correlations and then fixed-length blocks are made by using the effective bit reorganization algorithm. Finally, an effective bit allocation technique is used to limit error propagation in each block. Therefore, in low BER the proposed algorithm shows similar compression performance to the zerotree compression technique and in high BER it shows better performance in terms of PSNR than the conventional methods.

I. 서 론

최근 멀티미디어와 관련된 무선영상통신에 대한 요구와 인터넷 사용의 급증에 따라 웨이블릿 변환을 이용한 효율적인 영상 부호화 알고리즘이 활발히 연구되고 있다. 영상의 웨이블릿 부호화 과정은 영상의 상관도 제거, 부밴드 계수의 양자화, 양자화 계수의 엔트로피 부호화로 구성된다.[1] 최근에 소개된 제로트리 부호화 기법은 매우 높은 압축 성능을

* 삼성반도체연구원(jwyclub@nownuri.net) ** 연세대학교 전기컴퓨터공학과
논문번호 : 99236-0609, 접수일자 : 1999년 6월 9일

보이며, 효율적인 웨이블릿 영상 부호화 알고리즘을 제공한다.[2-4] 이러한 알고리즘은 주어진 임계치를 이용하여 웨이블릿 계수를 중요 계수와 비중요 계수로 분류하고, 비중요 계수를 부밴드간 트리관계를 이용하여 '제로트리'라는 하나의 부호로 부호화하여 압축 효율을 높일 수 있다. 따라서 부호화기에서 출력되는 비트열은 부호에 따른 계수의 주사 순서가 일정하지 일정하기 않기 때문에 잡음이 많은 이동통신 환경에서 비트 오류에 의해 영상을 심각하게 왜곡시킨다. 심지어 잡음에 의해 왜곡된 단 하나의 비트가 그 비트를 포함하는 계수나 집합뿐만 아니라 오류가 전파되어 영상 전체에 왜곡을 일으킬 수 있다. 따라서, 영상의 급격한 왜곡보다 BER(Bit Error Rate)의 증가에 따라 점차적인 성능의 감소가 영상의 완전복원이 필수적이지 않은 원천부호 시스템에서 중요하다. 이러한 문제를 해결하기 위한 일반적인 방법으로 동기화 부호를 보내는 방법이 있는데, 이것은 동기화 부호사이의 정보를 분리함으로써 오류의 확산을 제한하지만, 비교적 많은 양의 동기화 비트가 필요하여 비효율적인 부호화를 하게 된다.[5-6] 다른 방법으로 ECC(Error Correction Coding)와 FLC(fixed-length Coding)가 있다.[7-10] ECC는 부호에 추가적인 패리티 비트가 필요하여 압축 효율을 저하시키며 BER이 계속 증가하면 오류의 탐지 범위를 초과하여 영상의 급격한 왜곡이 생기게 되고, FLC는 효율적인 고정 길이 부호를 만들기 위한 추가적인 노력이 필요하지만, 압축 효율이 좋지 않다.

본 논문에서는 오류에 강인한 제로트리 웨이블릿 부호화 기법을 제안한다. 제로트리 웨이블릿 부호화 기법은 임의의 임계치에 대하여 중요 계수로 판별된 계수는 독립된 하나의 부호로 부호화하고, 비중요 계수로 판별된 다수의 계수들은 부밴드간 트리 구조를 이용하여 하나의 부호로 부호화하여 높은 압축 효과를 얻을 수 있다. 따라서 각 부호마다 부호화되는 계수의 개수가 다르며, 주사 순서는 부호에 의해 결정된다. 만약 잡음에 의하여 비트 오류가 발생하면 하나의 중요 계수는 다수의 비중요 계수로, 다수의 비중요 계수는 하나의 중요 계수로 잘못된 복호를 하게 된다. 따라서 부호화기와 복호화기의 주사 순서가 서로 다르게 되어 전체 영상으로 왜곡이 확대된다. 기존의 제안된 REZW(Robust Embedded Zerotree Wavelet) 부호화 기법[11]은 웨이블릿 계수들을 트리 구조를 이용하여 다수의 블록으로 분리한다. 그리고, 각 블록에 같은 양의 비트를 할당하여 제로트리 부호화 기법으로 부호화함으로써 블록의 동기화를 통해 비트 오류를 블록으로 제한하는 효과가 있다. 그러나, 경계 영역과 평활 영역에 같은 양의 비트를 할당하는 것은 부호화 효율을 크게 저하시키게 되어 오류가 없는 환경에서는 고화질의 영상을 제공하지 못하는 단점이 있다. 제안 알고리듬은 제로트리 부호화에서 생성된 비트열을 웨이블릿 계수에서의 트리 구조를 이용하여 가변 길이를 가지는 다수의 비트열로 분리하여 오류 비트에 의한 주사 순서의 붕괴를 블록으로 제한할 수 있다. 또한 가변 길이 블록들의 평균 길이를 계산하여 평균 길이보다 긴 블록의 비트열을 평균 길이보다 작은 블록으로 이동시킴으로써 같은 크기의 비트열로 재구성한다.[12] 따라서 블록의 시작점이 고정되므로 엔트로피 부호화에서 비트 오류에 의해 발생할 수 있는 블록의 동기 손실과 부호의 동기 손실을 해당 블록으로 제한하게 한다. 본 논문에서는 모의 실험을 통하여 기존의 부호화 기법과 객관적, 주관적 성능을 비교, 분석하여 제안 알고리듬의 우수성을 확인한다.

본 논문의 전체적인 구성은 다음과 같다. 제 2장에서는 배경 이론으로 제로트리 부호화 기법 중에 우수한 성능을 보이는 SPIHT 알고리듬[2]과 EREC 알고리듬[12]에 대하여 설명하고, 제 3장에서는 기존의 오류에 강인한 부호화 기법인 REZW 알고리듬[11]을 간단히 설명한다. 제 4장에서는 기존 알고리듬의 문제점을 분석하고 이를 해결할 수 있는 방식을 제안한다. 제 5장에서는 제안 방식의 시뮬레이션을 통하여 기존의 알고리듬들과 성능을 비교, 분석하며, 마지막으로 제 6장에서는 결론 및 향후 연구 과제에 대하여 기술한다.

Ⅱ. 배경 이론

1. SPIHT(Set Partitioning in Hierarchical Trees)

SPIHT[2] 알고리듬은 웨이블릿 계수들의 계층적 구조를 이용하여 식 (1)에서 2^n을 임계값으로 하여 분류 과정(sorting pass)과 세분화 과정(refinement pass)에 의해 부호화를 하게된다.

$$n = \lfloor \log_2(\max_{(i,j)}\{|C_{i,j}|\}) \rfloor \quad (1)$$

여기서, $C_{i,j}$는 좌표 (i,j)에서의 웨이블릿 계수이다. 웨이블릿 계수들은 이 임계값보다 크면 중요 계

수(significant coefficient)라 하고, 작으면 비중요 계수(insignificant coefficient)라 한다. 웨이블릿 계수들은 그림 1과 같이 부밴드 사이의 공간적인 트리 구조를 구성할 수 있다. 이러한 구조를 가지는 웨이블릿 계수는 일반적으로 에너지의 대부분이 저주파 부밴드에 집중되지만, 분산은 고주파 부밴드에서 저주파 부밴드로 갈수록 낮아진다. 또한, 부밴드의 같은 공간적인 상관도를 이용하여 하위 부밴드의 웨이블릿 계수를 예측할 수 있다.

SPIHT 알고리듬은 세 개의 목록을 사용하여 중요 계수와 비중요 계수를 나타내기 위해 비트를 할당한다. 트리루트(tree root)만 포함하는 비중요 화소의 목록(LIP, List of Insignificant Pixels), 자식 계수와 트리루트를 포함하는 비중요 집합의 목록(LIS, List of Insignificant Sets), 중요 계수를 의미하는 중요 화소의 목록(LSP, List of Significant Pixels)을 말한다. SPIHT 알고리듬의 분류 과정에서 비중요 화소의 목록의 계수들을 임계값과 비교하여 그 결과를 1비트로 출력한다. 만약 임계값보다 크다면 그 계수의 부호를 1비트로 출력하고, 비중요 화소의 목록에서 중요 화소의 목록으로 웨이블릿 계수를 옮긴다. 다음으로 비중요 집합의 목록에 속하는 모든 웨이블릿 계수들을 임계값과 비교하여 목록에 속하는 모든 웨이블릿 계수가 비중요 계수일 경우와 중요 계수가 한 개 이상 있을 경우로 나누어 1비트의 결과로 출력한다. 만약 한 개 이상 중요 계수가 있으면 부집합으로 나뉘어서 다시 임계값과 비교하여 1비트의 결과를 출력한다. 만약 임계값보다 크다면 그 계수의 부호를 1비트로 출력하고, 중요 계수의 목록으로 옮긴다. 마지막으로 이전의 임계값에 의해 결정된 중요 계수의 목록으로부터 현재의 임계값에 대한 비트맵이 출력된다. 분류 과정과 세분화 과정이 끝나면 임계값을 낮추고, 그 임계값에 대하여 다시 분류 과정과 세분화 과정을 하게 된다. 또한 더 효율적인 압축을 하기 위해 산술 부호화가 임계값에 대한 중요 또는 비중요 계수로 판별되는 비트열에 적용할 수 있다. 여기서 부호나 세분화 과정에서 생성되는 비트열에 산술 부호화를 적용하지 않는 이유는 0와 1의 발생 확률이 같이 때문에 계산적 복잡도만 증가시키고 압축 효율을 기대하기 어렵기 때문이다.

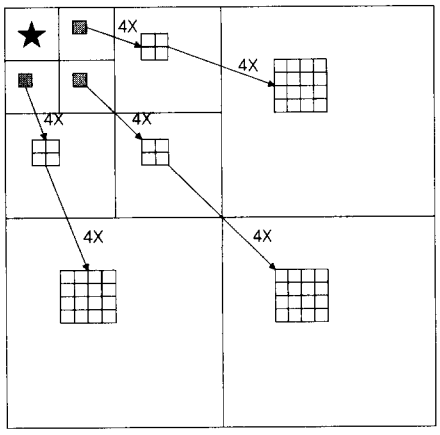


그림 1. 웨이블릿 변환 계수들의 트리구조

이러한 방식으로 생성된 비트열은 순차적 전송을 위한 임베디드(embedded) 부호화, 정확한 비트율 조절, 실시간 처리, 다해상도 스케일러빌리티(scalability)에 유리한 장점을 가지고 있다.

2. EREC(Error-Resilient Entropy Code)

대부분의 블록 기반 영상 부호화 기법은 블록마다 가변 비트열로 부호화되므로 잡음에 의한 비트 오류는 부호와 블록의 동기 손실을 유발하여 그 뒤를 따르는 부호와 블록의 시작점이 틀려지게 된다. EREC의 비트 재구성 알고리듬[12]은 가변 길이의 블록을 적은 잉여분으로 같은 길이의 블록으로 만들어서 블록의 시작점을 일정한 간격으로 고정하므로 블록의 재동기를 할 수 있게 된다.

EREC의 비트 재구성 알고리듬은 그림 2의 예와 같다. 그림 2는 각 블록의 비트열 길이 b_i ($i=1,2,\ldots,6$)가 11, 9, 4, 3, 9, 6비트를 가지는 6개의 블록을 평균 비트열 길이 s를 갖도록 재구성하는 알고리듬의 예를 보여주고 있다. 각 블록은 블록에 속하는 계수의 특성에 따라 중요도가 다르므로 할당되는 비트는 가변 길이를 가진다. 따라서 블록의 길이는 b_i와 s가 같을 경우, b_i가 s보다 작을 경우, b_i가 s보다 클 경우로 나누어 볼 수 있다. 그림 2에서는 블록 1, 2, 5가 b_i가 s보다 클 경우이고, 블록 3, 4, 6은 b_i가 s보다 작을 경우이다. b_i와 s가 같을 경우는 평균 비트량을 완전히 사용한 경우이고, b_i가 s보다 작을 경우는 평균 비트량을 완전히 사용하지 않아 $s-b_i$개의 비트가 남아있고, b_i가 s보다 클 경우는 b_i-s개의 비트가 평균 비트량보다 더 많이 사용되었다. 따라서 고정된 길이를 가지는 블록을 만들기 위해 평균 비트량보다 많이 사용된 블록의 비트를 남아있는 블록의 비트를 찾아서 삽입해야 한다. 즉, 각 단계 n에서 블록 i가 $i+\phi_n \pmod n$의 블록을 찾아서 이용할

수 있는 비트가 있다면 더 많이 사용된 비트의 전체 또는 일부를 그 블록에 삽입한다. 여기서 ϕ_n은 미리 정해진 오프셋 시퀀스(offset sequence)로 단방향 탐색, 양방향 탐색, 랜덤 탐색과 같이 세가지 경우가 있다. 그림 2에서는 단방향 탐색으로 오프셋 시퀀스가 0, 1, 2, 3, 4, 5인 경우이다. 일반적으로 의사랜덤(pseudo-random) 탐색이 n에 대하여 $\phi_{n+k}-\phi_n$이 독립이고, 두 블록간 k만큼 떨어져 있어서 같은 단계에서 탐색하지 않으므로 더 효율적으로 블록을 찾을 수 있다. 이와 같은 과정은 모든 블록이 s와 같을 때까지 계속하게 되고, 최종적인 비트열은 그림 2(d)와 같이 된다.

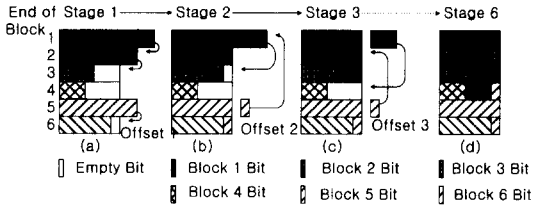


그림 2. 비트 재구성 알고리듬

Ⅲ. REZW(Robust Embedded Zerotree Wavelet)

REZW[11] 알고리듬의 기본 개념은 그림 3과 같다. REZW 알고리듬은 웨이블릿 계수를 트리 구조를 이용하여 S개의 블록으로 분리하고, 각 블록에 같은 크기의 비트를 할당하여 독립적인 S개의 제로트리 웨이블릿 알고리듬(SPIHT, EZW)으로 양자화, 부호화한다.

웨이블릿 계수는 부밴드간의 상호 관계를 이용하여 그림 4과 같이 두가지 방법으로 분리할 수 있다. 그림 4(a)는 식 (2)와 같이 부모 계수와 같은 공간적 위치에 있는 연속되는 4개의 자식 계수로 분리하는 제로트리 보존 방법이고, 그림 4(b)는 식 (3)과 같이 부모 계수와 같은 공간적 위치에 있는 1개의 자식 계수와 오프셋(offset)만큼 떨어진 계수로 분리하는 오프셋 제로트리 분리 방법이다.

$$O(i,j) = \{(2i,2j),(2i,2j+1),(2i+1,2j),(2i+1,2j+1)\} \quad (2)$$

$$O(i,j) = \{(2i,2j),(2i,2j+offset),(2i+offset,2j),(2i+offset,2j+offset)\} \quad (3)$$

여기서, $O(i,j)$는 좌표 (i,j)에서의 모든 자식 계수의 집합이고, $offset$은 2이상의 정수이다. 웨이블릿 계수의 부밴드간 상관도를 이용한제로트리 보존 방법이 압축률은 더 좋으나 오류가 증가하면 오프셋 제로트리 분리 방법이 밴드간 상관도가 낮기 때문에 오류의 전파가 작아져서 오류에 더 강인하게 된다.

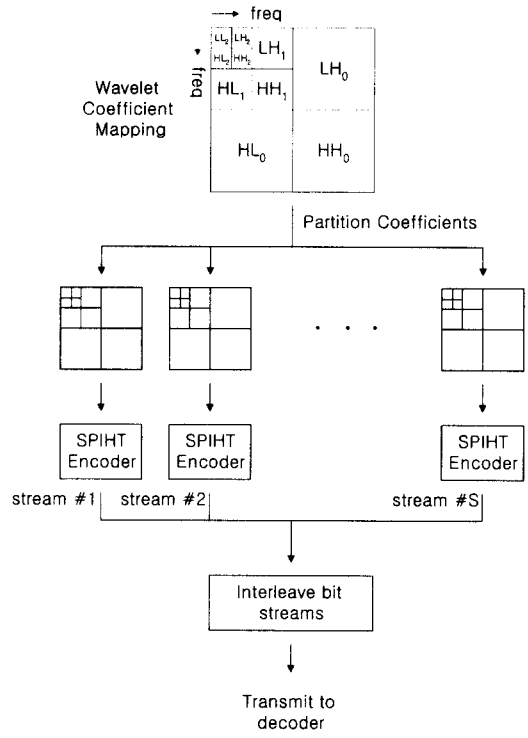


그림 3. REZW 알고리즘의 구조

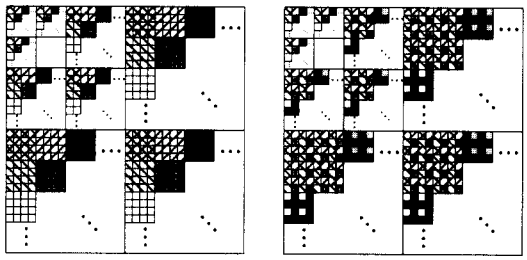


그림 4. 웨이블릿 계수 분리
(a) 제로트리 보존 분리 (b) 오프셋 제로트리 분리

최종적으로 비트열은 전송하기 전에 알맞은 크기(비트, 바이트, 패킷)로 인터리브(interleave) 시켜서 복호화기로 전송하게 된다. 이렇게 조합한 비트열은

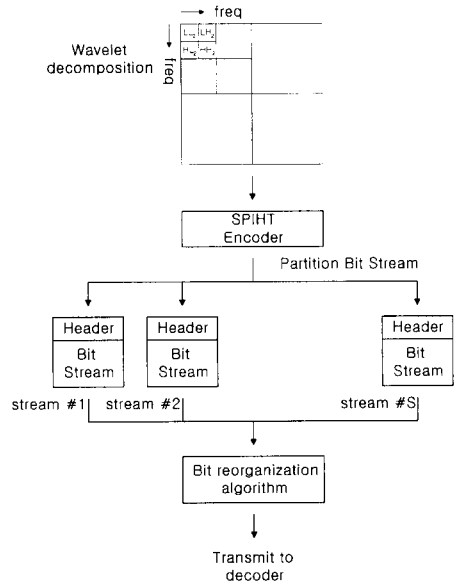
대역폭에 따라 압축률을 선택할 수 있는 임베디드 부호화의 특징을 유지한다. 또한, 각 블록에 일정한 비트를 할당하여 고정 간격으로 블록이 시작되므로 복호화기에서는 오류에 의한 블록 동기의 손실 및 부호 동기의 손실을 다음 블록에서 복구 할 수 있다.

Ⅳ. 제로트리 웨이블릿에서 오류에 강인한 영상 압축

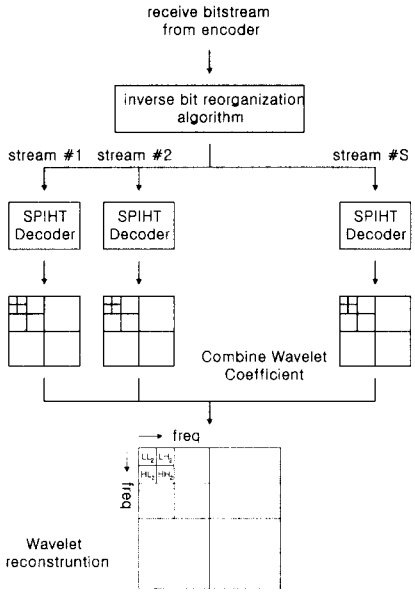
REZW 알고리듬[11]은 일반적인 제로트리 웨이블릿 알고리듬[2-4]과 비교해 보았을 때, 잡음이 많은 환경에서는 비트 오류에 의한 주사 순서의 붕괴 및 블록과 부호의 동기 손실을 블록으로 제한한다. 하지만, 오류에 강인한 부호화를 위하여 모든 블록에 같은 양의 비트를 할당하므로 중요한 영상 정보인 경계 영역이 속하는 블록에는 많은 비트가 할당되지 못하고, 중요하지 않은 평활 영역이 속하는 블록에는 많은 비트가 할당하게 된다. 그로 인하여 오류가 없거나 적은 환경에서는 고화질의 영상을 제공하지 못할 뿐만 아니라 블록마다 화질이 고르지 못한 단점이 있다. 본 논문에서는 이와 같은 문제점을 해결할 수 있도록 제로트리 웨이블릿 알고리듬으로 생성된 비트열을 다수의 블록으로 분리하여 효율적인 비트 할당을 하게 하고, 비트 재구성 알고리듬을 이용하여 오류에 강인한 부호화 기법을 제안한다.

그림 5는 제안 알고리듬의 전체 구성도이다. 그림 5(a)의 부호화기에서는 웨이블릿 변환을 한 후, 계수를 분리하여 각 블록마다 제로트리 부호화 기법으로 부호화 하는 것이 아니라 먼저 제로트리 부호화 기법으로 부호화 한 후, 그림 6과 같이 비트열을 블록으로 분리한다. 그림에서 각각의 알파벳은 다른 트리에서 부호화된 부호를 의미하며, 알파벳 오른쪽의 숫자는 하나의 트리에서 생성되는 부호의 순서이다. 비트열을 분리하기 위해서는 제로트리 부호화기에서 부호가 생성될 때, 각각의 부호가 어느 위치의 계수로부터의 부호화 되었는지 알 수 있어야 되므로 꼬리표로 좌표가 같이 출력되어야 한다. 이 꼬리표는 비트열을 정렬하고 분리하는 과정에서만 사용이 되고 복호화기로 전송하지 않는다. 제로트리 부호화 기법에서 생성된 비트열을 꼬리표에 나타난 좌표를 이용하여 같은 트리구조에 속하는 계수의 부호화 비트를 생성된 순서로 정렬하고, 비트열을 분리한다. 이 과정으로 각 블록에서 중요 계수를 블록의 기준에서 판별하여 찾는 것이 아니라 전체를 기준으로 판별되므로 효율적인 부호화를 할 수 있게 한다. 분리된 비트열은 각 블록마다 가변 길이 부호를 가지는 산술 부호화(arithmetic coding)를 이용하여 추가적인 압축을 한다. 각 블록은 비트열을 정렬과 분리, 산술 부호화를 적용하여 가변 길이를 가지는 블록이므로 각 블록의 길이를 헤더 정보로 기록한다. 하지만, 헤더 정보에 비트 오류가 발생하면 복호화기에서 잘못된 역 비트 재구성을 하게 되어 영상 화질의 왜곡이 심하게 되므로, 헤더 정보는 오류가 가장 심한 경우에서 오류를 탐지하고 수정할 수 있는 FEC(Forward Error Correction) 부호화기를 이용하여 보호한다. 이러한 헤더 정보의 크기는 전체 비트열에서 0.01BPP(Bit Per Pixel) 정도가 소모되며, 이 정도 비트 소모는 오류가 없는 환경에서 최적화된 제로트리 부호화 기법의 결과와 PSNR로 비교해 보았을 때, 거의 차이가 나지 않는다.

다음 과정으로 분리된 가변 길이의 블록을 비트 재구성 알고리듬으로 같은 크기의 블록으로 재구성하여 일정한 간격 후에 블록의 재동기를 할 수 있게 한다. 이 알고리듬은 EREC의 비트 재구성 알고리듬[12]과 비슷하지만, EOB(End of Block)를 나타내는 방법에서 차이가 있다. EREC의 비트 재구성 알고리듬에서는 허프만 부호화(Huffman coding)를 이용하고, 각 블록에서 EOB를 나타내는 부호를 추가하게 되는데, 제안 알고리듬에서는 산술 부호화를 이용하고, EOB를 위한 새로운 부호의 추가는 부호



(a)



(b)

그림 5. 제안 알고리듬의 구조 (a) 부호화기 (b) 복호화기

의 비트량을 증가시키므로 비효율적이다. 그러므로, EOB를 나타내는 새로운 부호 대신에 그 위치 정보를 추가하여 표시하는 것이 더 효율적이다. 제안 알고리듬에서는 비트열을 정렬하고 분리할 때, 가변 길이의 블록을 나타내기 위하여 헤더 정보를 추가하였는데, 이를 이용하면 추가적인 정보의 전송을 하지 않고 EOB를 복호화기에서 알 수 있으므로 역 비트 재구성 알고리듬을 적용할 수 있다.

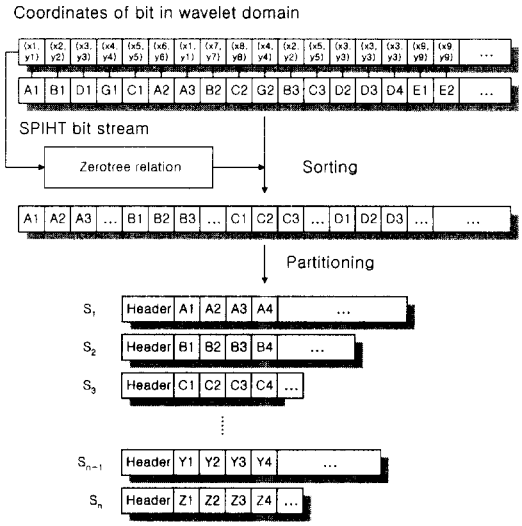


그림 6. 비트열의 정렬과 분리

그림 7은 제안 알고리듬의 복호화기에서 사용된 역 비트 재구성 알고리듬의 예이다. 헤더 정보를 통해 블록 1, 2, 5가 블록의 평균 길이보다 크고, 블록 3, 4, 6이 블록의 평균 길이보다 작으므로 블록 1, 2, 5의 비트열이 블록 3, 4, 6의 비트열에 포함되었음을 알 수 있다. 그림에서 단방향 탐색을 이용하였다고 가정하면 그림 7(a)에서 블록 1은 블록 2, 블록 2는 블록 3, 블록 5는 블록 6을 탐색한다. 헤더 정보를 통해 블록 1이 탐색한 블록 2는 평균 블록보다 큰 블록이므로, 비트열을 삽입하지 않고, 블록 2가 탐색한 블록 3은 평균 블록보다 작은 블록이므로 블록 3의 헤더 정보에 나타난 블록의 위치 이후로부터 블록 2의 헤더 정보에 나타난 블록의 크기를 만족할 때까지 비트열을 삽입한다. 블록 5가 탐색한 블록 6도 같은 방법으로 비트열을 삽입해야 하지만, 블록 5의 헤더 정보에 나타난 블록의 크기를 만족하지 못했기 때문에 다음 단계에서도 계속 탐색하게 된다. 그림 7(b)에서는 블록 1, 5가 헤더 정보에 나타난 블록의 크기를 만족하지 못했으므로 블록 1은 블록 3, 블록 5는 블록 1을 탐색한다. 블록 1이 탐색한 블록 3은 평균 블록보다 작은 블록이고 그림 7(a)에서 블록 2가 삽입하고 남은 비트열이 있기 때문에 블록 1에 삽입한다. 하지만, 블록 1의 헤더 정보에 나타난 블록의 크기를 만족하지 못했기 때문에 다음 단계에서 탐색을 계속하게 된다. 또한 블록 5가 탐색한 블록 1은 평균 비트열보다 큰 블록이므로, 비트열을 삽입하지 않는다. 그림 7(c)에서는 블록 1이 블록 4, 블록 5가 블록 2를 탐색한다. 블록 1이 탐색한 블록 4는 평균 비트열보다 작은 블록이므로 블록 4의 헤더 정보에 나타난 블록의 위치 이후로부터 블록 1의 헤더 정보에 나타난 블록의 크기를 만족할 때까지 비트열을 삽입한다. 블록 5가 탐색한 블록 2는 평균 비트열보다 큰 블록이므로 블록 5만 다음 단계로 넘어가게 된다. 이 과정을 그림 7(d)까지 계속하면 최종적으로 부호화기에서 비트 재구성 알고리듬을 사용하기 전의 가변 길이를 가지는 블록으로 역 재구성 할 수 있다.

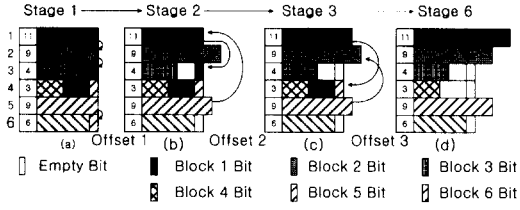
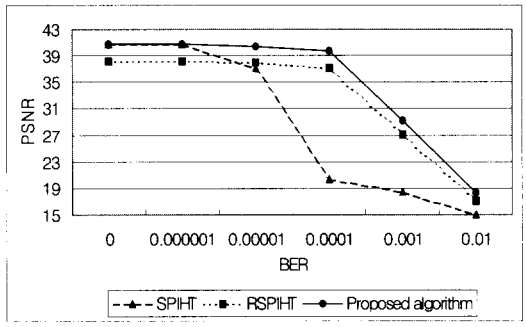


그림 7. 역 비트 재구성 알고리듬

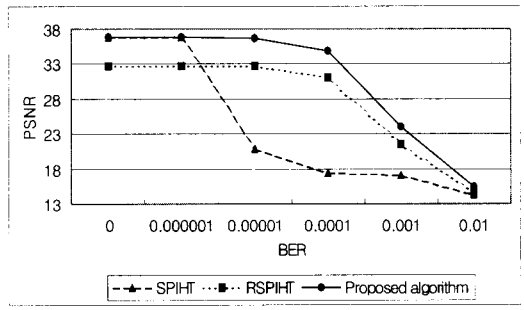
그림 5(b)는 제안 알고리듬의 복호화기이다. 먼저, 수신된 비트열은 그림 7과 같이 역 비트 재구성 알고리듬을 적용하여 고정된 길이의 블록을 가변 길이의 블록으로 만든다. 이 과정으로 각 블록은 한 트리에 속하는 비트열끼리 다시 모이게 된다. 그리고 각 블록에 제로트리 알고리듬 복호화기를 병렬로 적용하여 하나의 트리에 대한 웨이블릿 계수를 복원한다. 이렇게 병렬로 복호화된 계수를 블록의 순서와 트리의 구조를 이용하여 전체 웨이블릿 계수로 조합하고, 마지막으로 역 웨이블릿 변환을 적용하여 영상을 복원하게 된다.

V. 시뮬레이션 결과 및 고찰

제안 알고리듬은 8비트 해상도를 갖는 512×512 영상을 9/7 쌍직교(biorthogonal) 필터로 5-레벨까지 웨이블릿 변환하여 모의 실험을 하였다. 제로트리 부호화 기법으로는 SPIHT 알고리듬을 이용하여 REZW 알고리듬과 제안 알고리듬을 구현하였다. 따라서 REZW 알고리듬은 RSPIHT(Robust Set Partitioning in Hierarchical Trees) 알고리듬이라고 하겠다. SPIHT, RSPIHT 알고리듬과 제안 알고리듬을 같은 압축률로 부호화하여 BSC(Binary Symmetric Channel) 환경에서 오류를 증가시키면서 결과를 비교하였다. 제로트리 압축 알고리듬은 오류의 위치에 따라 성능이 크게 좌우되기 때문에 SPIHT, RSPIHT 알고리듬과 제안 알고리듬은 같은 위치에 오류를 삽입하였다. 또한 RSPIHT 알고리듬은 제로트리 보존 방법으로 웨이블릿 계수를 분리를 하였으며, 제안 알고리듬과 RSPIHT 알고리듬의 정확한 비교를 위해서 같은 개수로 블록을 분해하였다. 그림 8은 1BPP로 Lena 영상과 Barbara 영상을 부호화 하였을 때, BER의 변화에 따른 PSNR을 그래프로 나타내었다. 그림 8에서와 같이 제안 알고리듬은 오류에 강인한 부호화를 위해서 RSPIHT보다 적은 비용을 소모하여 오류가 없을 때는 SPIHT와 유사한 성능을 나타낸다. 또한 BER의 증가에 따라 SPIHT뿐만 아니라 RSPIHT와 비교하였을 때, 오류에 더 강인한 성능을 보여주고 있다. 그림 9는 주관적인 평가를 위해 1BPP로 Barbara 영상을 부호화하고, BER이 10^{-4}일 때 성능을 비교하였다. 그림 9(a)는 원영상이고, 그림 9(b)는 SPIHT 알고리듬으로 부호화 한 것으로 오류가 영상 전체로 전파되어서 매우 많이 왜곡되었음을 알 수 있다. 그림 9(c)는 RSPIHT 알고리듬으로 부호화 한 것으로 오류의 전파를 트리 블록으로 제안하였지만, 전체적인 영상의 화질이 저하되고 여러 곳에서 영상이 왜곡되었음을 확인할 수 있다. 그림 9(d)는 제안 알고리듬으로 부호화 한 것으로 RSPIHT 알고리듬에 비해 향상된 화질을 보이고 있고, 오류를 트리 블록으로 제안하였을 뿐만 아니라 왜곡된 정도도 RSPIHT 알고리듬보다 적음을 알 수 있다. 따라서 제안 알고리듬은 기존의 오류에 강인한 부호화보다 적은 비용을 사용하여 오류에 더 강인한 부호화 기법임을 객관적, 주관적으로 비교하여 확인되었다.

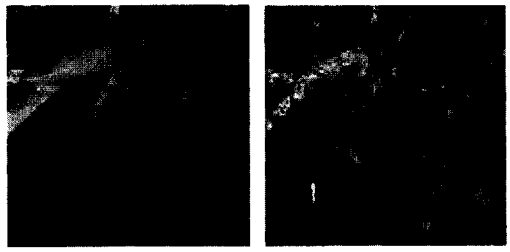


(a)

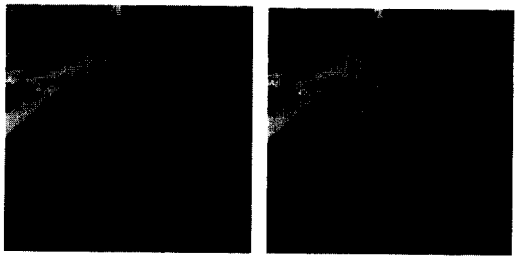


(b)

그림 8. 1BPP에서의 BER의 변화에 따른 (a) Lena와 (b) Barbara 영상의 PSNR



(a) (b)



(c) (d)

그림 9. 시뮬레이션 결과 영상 (1BPP, BER=10^{-4})
(a) 원영상, (b) SPIHT, (c) RSPIHT,
(d) 제안 알고리듬

Ⅵ. 결 론

본 논문에서는 제로트리 웨이블릿 부호화 방식을 이용하여 오류에 강인한 부호화 기법을 제안하였다. 웨이블릿 계수 대신 부호화된 비트열을 분리하여 부호화 효율을 높였으며, 비트 재구성 알고리듬으로 부호 및 블록의 시작점을 일정하게 하고, 오류에 의한 주사 순서의 붕괴를 블록으로 제한하였다. 시뮬레이션 결과 효율적인 비트 할당으로 기존의 오류에 강인한 부호화 기법보다 고화질의 영상을 제공하며, 더 오류에 강인함을 객관적, 주관적으로 확인할 수 있었다. 또한 전체적인 부호화 알고리듬의 수정을 하지않고, 비트의 정렬 및 분리 과정과 비트 재구성 알고리듬만 추가하면 간단하게 구현할 수 있고, 제로트리 부호화 기법을 사용하는 EZW, SPIHT, SFQ 알고리듬에서 같은 방식으로 적용시킬 수 있는 장점을 가지고 있다. 향후 연구 과제로는 불가피하게 발생된 오류를 복구해 줄 수 있도록 오류를 발생을 탐지하고 이를 복구할 수 있는 알고리듬을 연구할 필요가 있으며, 웨이블릿 동영상 부호화에서 오류에 강인한 부호화 기법의 연구가 수행되어야 하겠다.

참 고 문 헌

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

[2] 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.

[3] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," IEEE trans. Signal Processing, vol. 41, pp. 3445-3462, Dec. 1993.

[4] Z. Xiong, K. Ramchandran, and M. Orchard, "Space-frequency quantization for wavelet image coding," IEEE Trans. Image Processing, vol. 6, pp. 677-693, May 1997

[5] T. J. Ferguson and J. H. Rabinowitz, "Self-synchronizing Huffman codes," IEEE Trans. Inform. Theory, vol. 30, pp. 687-693, July 1984.

[6] B. L. Montgomery and J. Abrahams, "Synchronizing of binary source codes," IEEE Trans. Inform. Theory, vol. IT-32, pp. 849-854, November 1986.

[7] N. T. Cheng and N. G. Kingsbury, "The ERPC: An Efficient Error-Resilient Technique for Encoding Positional Information or Sparse Data," IEEE Trans. Communications, vol. 40, pp. 140-148, January 1992.

[8] J. W. Modestino and D. G. Daut, "Combined source-channel coding of images," IEEE Trans. Communications, vol. 27, pp. 1644-1659, November 1979.

[9] N. Farvardin, "A study of vector quantization for noisy channels," IEEE Trans. Inform. Theory, vol. 36, pp. 799-809, July 1990.

[10] R. Llados-Bernaus and R. L. Stevenson, "Fixed-Length Entropy Coding for Robust Video Compression," IEEE Trans. Circuits and Systems for Video Technology, vol. 8, pp. 745-755, October 1998.

[11] C. D. Creusere, "A new method of robust image compression based on the embedded zerotree wavelet algorithm," IEEE Trans. Image Processing, vol. 6, pp. 1436-1442, October 1997.

[12] D. W. Redmill and N. G. Kingsbury, "The EREC: An Error-Resilient Technique for Coding Variable-Length Blocks of Data," IEEE Trans. Image Processing, vol. 5, pp. 565-574, April, 1996.

장 우 영(Woo Young Jang) 준회원
1998년 2월 : 경희대학교 전자공학과 졸업(공학사)
2000년 2월 : 연세대학교 대학원 전기컴퓨터공학과 졸업(석사)
2000년 2월~현재 : 삼성반도체 연구원
<주관심 분야> 웨이블릿 기반 영상부호화

송 환 종(Hwang Jong Song)
1996년 2월 : 연세대학교 전파공학과 졸업(공학사)
1998년 2월 : 연세대학교 대학원 전기컴퓨터공학과 졸업(석사)
1998년 2월~현재 : 연세대학교 대학원 전기컴퓨터 공학과 박사과정
<주관심 분야> 웨이블릿 기반 영상부호화, 비디오 영상 처리 및 압축

손 광 훈(Kwanghoon Sohn) 정회원
한국통신학회 논문지 제 24권 7B호 참조
현 재 : 연세대학교 전기컴퓨터공학과 교수