

블록 기반 제로트리와 쿼드트리를 이용한 정지영상 부호화

김효준, 이상욱

서울대학교 전기공학부 뉴미디어통신공동연구소 영상처리연구실

Still image coding using block-based zerotree and quadtree

Hyo Joon Kim and Sang Uk Lee

Image Processing Lab., Inst. of New Media and Comm., Seoul Nat'l Univ.

hades@iclsnu.ac.kr

요 약

웨이블렛 변환을 이용한 영상 압축의 방법에 있어서 최근 우수한 압축 성능을 보이는 비트 평면 부호화 방법이 많이 발전되어 왔다. 본 논문에서는 블록 기반 제로트리와 사진나무구조를 이용하여 비트 평면을 효율적으로 부호화함으로써 우수한 압축 성능을 보이는 부호기를 제안한다. 블록 기반 제로트리는 주파수 대역에 걸쳐 상관성을 보이며 존재하는 중요하지 않은 계수를 효과적으로 부호화하며, 사진나무구조는 블록 안에 있는 중요한 계수를 찾아내어 부호화한다. 제안하는 부호기의 성능은 기존의 우수한 비트 평면 부호기에 필적하는 성능을 보이며, 특히 낮은 피라미드 레벨과 저비트율에서의 성능이 더 우수함을 보인다.

1. 서론

근래 웨이블렛(wavelet) 이론이 소개되면서 웨이블렛 변환에 기반한 영상 부호화에 대한 연구가 많이 이루어지고 있다. 이것은 웨이블렛 변환이 에너지 압축과 공간-주파수 국지화와 같은 여러 유용한 특성을 제공함에 기인한 것이다. 최근 Shapiro에 의해 제안된 EZW[1] 방법은 공간-주파수 국지화, 주파수 대역간의 유사성, 주파수 대역에 따른 웨이블렛 계수의 크기 감소 등과 같은 웨이블렛 변환 후의 영상의 특성을 이용하여 효율성과 복잡도 면에서 두 가지 측면을 동시에

향상시키는 매우 우수한 영상 압축 성능을 보였다. Said와 Pearlman에 의해 제안된 SPIHT[2] 방법은 중요도 지도(significance map)를 부호화하는데 새로운 집합 분할 방법을 사용하여, 산술 부호기를 사용하지 않고도 EZW보다 우수한 성능을 보였다. EZW와 SPIHT 두 방법 모두 중요도 지도를 부호화하기 위해 구조화된 계수 집합인 제로트리(zerotree)를 사용하여 중요하지 않은 웨이블렛 계수들이 주파수 대역간에 갖고 있는 상관성을 이용한다. 한편 Servetto 등에 의해 제안된 MRWD[3] 방법과 Chai 등에 의해 제안된 SLCCA[4] 방법은 주파수 부대역(subband) 안에 중요한 웨이블렛 계수들이 군집형태를 이루는 특성과 부대역간에 중요한 계수들이 서로 관련되는 특성을 이용하여 중요도 지도를 효율적으로 부호화를 함으로써, 제로트리에 기반한 방법들에 필적하는 성능을 보였다.

위와 같은 두 종류의 접근방법은 웨이블렛 변환된 영상의 특성을 이용하여 중요도 지도를 효율적으로 부호화하는 것을 공통의 목표로 갖고 있지만 서로 다른 측면에서 알고리즘을 발전시켰다. 즉, 위에 언급된 부호화 방법들은 주어진 비트 평면(bit plane)에서 웨이블렛 계수가 중요한지 아닌지를 최대한 압축하여 기술하는데 주안점을 두고 있는데, EZW와 SPIHT과 같은 제로트리에 기반한 방법들은 제로트리의 형태로 영인 계수(중요하지 않은 계수)를 효과적으로 부호화를 한 후 계수가 영인 영역들을 배제함으로써 영이 아닌 계

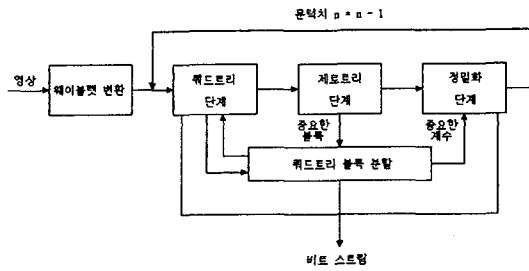


그림 1 제안 알고리즘의 기본적인 블록도

수들의 위치를 파악하고, MRWD와 SLCCA 방법은 군집하여 모여 있는 영이 아닌 계수들을 둘러싸는 계수들만을 부호화하여 그 안에 있는 계수들을 중요한 계수라고 파악하고 그 외의 영역들은 중요하지 않은 영역으로 파악한다.

이 논문에서는 블록 기반 제로트리(zerotree)와 쿼드트리(quadtree)를 이용하여 중요도 지도를 효율적으로 부호화 하는 방법을 제안한다. 본 논문에서 소개되는 블록 기반 제로트리는 계층적인 나무 구조의 각각의 노드를 구성하는 것이 웨이블릿 계수가 아니라 여러 계수들을 포함하고 있는 블록인 구조를 갖는다. 제안하는 알고리즘은 SPIHT과 비슷한 방법으로 넓은 범위에 걸쳐있는 영인 계수들을 블록기반 제로트리에 의하여 부호화하고 중요한 계수를 포함하고 있는 블록을 찾아낸다. 제로트리 부호화 과정에서 찾은 중요한 블록은 사진나무구조 방법에 의하여 블록을 단계적으로 분할하여 블록 안에 위치한 중요한 계수를 찾아낸다. 즉, 제안하는 부호화 알고리즘은 제로트리 방법이 중요하지 않은 계수들을 효과적으로 부호화하는 특성을 이용하고, 중요한 블록을 쿼드트리 방법으로 분할하여 일반적으로 군집형태로 존재하는 중요한 계수들을 효과적으로 찾아 부호화하여 효율적으로 중요도 지도를 부호화하게 되는 것이다.

2. 제안 부호화 알고리즘

그림 1은 제안하는 알고리즘의 기본적인 블록 구조도를 보인다. 제안 알고리즘은 제로트리, 쿼드트리, 정밀화의 세 단계로 이루어진다. 제로트리 단계에서는 제

안 알고리즘은 공간 방향 나무구조(spatial orientation tree)를 따라 블록들을 중요도 테스트를 함으로써 웨이블릿 계수들의 블록들을 정렬하고, 결국 중요한 블록을 찾는다.

k 는 음수가 아닌 정수일 때, 블록은 $2^k \times 2^k$ 크기의 웨이블릿 계수를 담고 있다. 만약 어떤 블록 B 가 n 에 대하여 다음과 같은 식을 만족한다면 그 블록은 중요하다고 정의된다.

$$\max_{(i,j) \in B} |c_{i,j}| \geq 2^n$$

위 식에서 $c_{i,j}$ 는 웨이블릿 계수이고, (i,j) 는 그 계수의 좌표를 나타낸다. 만약 주어진 블록이 위 식을 만족하지 않으면 그 블록은 중요하지 않다고 말한다. 공간 방향 나무구조는 웨이블릿 변환된 영상의 대역간 유사성과 계수의 국지화를 이용하여 블록 그룹의 중요도를 평가하기 위해 도입된 것으로, 공간 방향 나무구조에서는 각 노드는 2 x 2개의 인접한 블록들로 이루어져 있고, 노드의 각 블록은 4개의 자식 블록을 갖는다. 단 웨이블릿 피라미드의 최상위 레벨에서는 노드안의 하나의 블록만은 아무 자식 블록도 갖지 않는다. 실제적으로 알고리즘을 구현할 때, 제안 알고리즘은 4개의 목록을 유지하게 되는데, 그 목록들은 각각 중요하지 않은 블록 목록(LIB), 중요하지 않은 블록 집합 목록(LIBS), 중요한 계수 목록(LSP), 쿼드트리 적용 블록 목록(LQB)이다. 초기화 단계에서는 제안 알고리즘은 LIB를 웨이블릿 피라미드의 최상위 레벨에 있는 모든 $2^i \times 2^i$ 크기의 블록들로(i 는 양의 정수), LIBS를 자손을 갖지않는 블록을 제외한 최상위 레벨에 있는 모든 블록들로 초기화하고, LSP와 LQB는 비어있는 목록으로 설정한다. 실제적으로 제로트리 단계에서 하는 기본적인 작용은 LIBS 상에 있는 블록 집합을 반복적으로 분할하여 각각의 중요하지 않은 블록과 전보다 더 작은 크기의 중요하지 않은 블록 집합을 찾아서 각각의 위치를 LIB와 LIBS에 알맞게 등재시키고, 결국에는 중요한 블록을 찾아내는 것이다. 일단 중요한 블록이 찾아지면 그 블록에 즉시 쿼드트리 블록 분할 방법을 적용한다.

쿼드트리 블록 분할 방법은 문턱치 n 에 대하여 블록의 중요도를 평가하면서 그 블록에 쿼드트리 분할을 수행한다. 만약 주어진 블록이 중요하다면, 그 블록은 블록의 4분의 1 크기를 갖는 더 작은 4개의 블록으로 분할된다. 이 4개의 블록들은 각각 다시 주어진 블록으로 설정되어 화소 단위로 작아질 때까지 반복적으로 처리된다. 이런 반복적인 과정에 의하여 처음 블록 안에 존재하는 중요한 계수들이 찾아지고 부호화되며, 이렇게 찾아진 중요한 계수들은 LSP 상에 등재된다. 이 과정 중에 중요하지 않다고 판정된 블록들은 LQB 상에 등재되며 쿼드트리 단계에서 다음의 더 낮은 문턱치에 대하여 다시 중요도 평가를 받게 된다.

쿼드트리 단계에서는 LQB 상에 등재된 중요하지 않은 블록들을 현재의 문턱치에 대하여 쿼드트리 블록분할을 수행하게 되는데, LQB 상에는 다양한 크기를 갖는 블록, 즉 $2^k \times 2^k$ 크기의 블록(여기서 $k = 0, 1, \dots, i - 1$)이 존재할 수 있다. 이 블록들은 그 크기가 작은 순서대로 차례대로 처리를 한다. 즉 1×1 크기의 블록을 맨 먼저 처리를 하고 그 다음 2×2 크기의 블록들을 처리하는 순서이다. 크기가 작은 순으로 블록을 처리하는 이유는 LQB 상에 등재된 블록들은 모두 그 블록과 크기가 같은 인접 블록에 중요한 계수가 있을 경우에 생긴 블록들이다. 따라서 블록의 크기가 작을 수록 그 블록이 연속되는 문턱치에서 먼저 중요한 계수를 담고 있을 가능성이 높기 때문이다.

사진나무구조 단계 후의 정밀화 단계에서는 이 전의 문턱치에서 LSP 상에 등재된 계수들의 현재의 비트 평면상의 비트를 출력함으로써 먼저 발견된 중요한 계수들을 더 정확한 원래의 값으로 복원하게 한다. 이 세 단계는 현재의 문턱치를 연속적으로 2배 낮추면서 원하는 비트율이나 영상 화질이 달성될 때까지 반복된다. 이 때 각 단계에서 출력된 비트들은 엔트로피 부호기를 거치지 않고 바로 비트 스트림이 될 수도 있고, 엔트로피 부호기를 거쳐 비트율을 더욱 낮출 수 있다.

3. 실험결과

표 1 웨이블릿 피라미드 레벨에 따른 SPIHT과 제안 알고리즘의 성능 비교 (단위 : dB)

피라미드 레벨 \ 비트율 (bpp)		0.125	0.25	0.5	1.0
3 레벨 피라미드	SPIHT	23.55	26.46	30.74	36.32
	제안 알고리즘	24.62	27.69	31.71	37.03
5레벨 피라미드	SPIHT	24.95	27.72	31.65	36.93
	제안 알고리즘	25.46	28.22	32.03	37.22

실험 영상으로는 화소 당 8비트의 정밀도를 갖고, 512×512 크기의 흑백 영상인 Lena와 Barbara 영상을 사용하였고, 이 영상들은 9-7 Daubechies 필터[5]를 사용하여 웨이블릿 변환을 하였다. 비트율은 실제 압축된 파일의 실제 크기로부터 계산되었고, 영상 화질은 실제 복원된 영상으로부터 계산된 PSNR(peak signal to noise ratio)에 의해 측정되었다.

표 1은 Barbara 영상을 웨이블릿 변환할 때 분할되는 피라미드의 레벨을 달리한 후 8×8 블록을 사용한 제안 알고리즘과 SPIHT의 성능을 비교한 것이다. 두 가지 경우의 레벨에서 제안 알고리즘이 SPIHT보다 모두 우수한 결과를 보이는데, 피라미드 레벨이 5일 경우보다 피라미드 레벨이 3일 경우 제안하는 알고리즘이 SPIHT의 성능보다 더 우수한 결과를 보인다. 이는 SPIHT이 오로지 공간 방향 나무구조 상의 부모-자식 간의 관계를 이용한 제로트리만을 이용한 알고리즘인데 반해 제안 알고리즘은 제로트리 뿐만 아니라 쿼드트리 방법도 사용하기 때문이다. 피라미드 레벨이 작을 경우에는 공간 방향 나무구조를 이용하는 제로트리에 의해 얻을 수 있는 이득이 작기 때문에 이 관계만을 이용하는 SPIHT의 성능이 감소되지만, 제로트리 뿐만아니라 쿼드트리를 이용하는 제안 알고리즘은 피라미드 레벨이 낮은 경우에도 우수한 성능을 보이는 것이다.

표 2는 Barbara와 Lena 영상에 대하여 제안 알고리즘과 SPIHT 알고리즘의 비트율에 따른 복원된 영상의

표 2 제안 알고리즘과 SPIHT의 성능 평가 (단위 : dB)

실험영상		비트율 [bpp]			
		0.125	0.25	0.5	1.0
Barbara	SPIHT	23.55	26.46	30.74	36.32
	제안 알고리즘	24.62	27.69	31.71	37.03
Lena	SPIHT	27.53	32.01	36.12	39.66
	제안 알고리즘	29.72	33.21	36.62	39.90

화질을 비교한 것을 보여준다. 제안 알고리즘과 SPIHT 모두 산술부호기 없이 사용하였으며, 피라미드 레벨은 3이고, 제안 알고리즘의 초기 블록 크기는 8 x 8로 하였다. 모든 비트율에 있어서 제안 알고리즘이 SPIHT 보다 우수한 것을 볼 수 있으며, 특히 낮은 비트율에서는 더욱 더 성능이 우월함을 볼 수 있다. 이는 낮은 비트율일 때는 최상위 비트 평면 몇 개만을 부호화하게 되는데 이런 비트 평면에서는 중요하지 않은 계수들이 중요한 계수들보다 더 많이 존재하게 되고, 이런 경우 부호기는 중요하지 않은 계수들을 부호화하는데 많은 비트량을 소모하게 된다. 이때 제안 알고리즘은 하나의 계수가 아닌 여러 계수의 집합인 블록으로 중요하지 않은 계수들을 더 적은 수의 제로트리로 부호화를 할 수 있게되며, 이런 계수들을 부호화하는데 비트량을 감소할 수 있게된다. 감소한 만큼 중요한 계수들을 부호화하는데 비트를 더 사용할 수 있게되며, 결과적으로 복원되는 영상의 화질을 향상시키게 되는 것이다. 하지만 비트율이 높아질수록 비트 평면상의 중요한 계수의 비율이 많아지기 때문에 블록 기반 제로트리의 장점이 감소할 뿐만 아니라, 블록을 쿼드트리 분할하여 부호화하는데 비트량이 필요하기 때문에, 고비트율에서는 블록의 크기에 따른 성능차가 줄어들게 된다.

4. 결론

본 논문에서는 블록 기반 제로트리와 사진나무구조 블

록 분할을 이용하여 웨이블릿 변환된 영상의 중요도 지도를 효과적으로 부호화하는 알고리즘을 제안하였다. 본 논문에서 제안한 블록 기반 제로트리 방법은 넓은 범위에 걸쳐 있는 중요하지 않은 계수들을 효과적으로 부호화를 하며, 중요한 계수를 담고 있는 블록을 찾아낸다. 이 블록은 쿼드트리 분할방법으로 분할되어 블록 안에 위치한 중요한 계수들이 발견되어 부호화된다. 제안하는 알고리즘은 제로트리 뿐만 아니라 쿼드트리 분할 방법을 사용하기 때문에 낮은 레벨의 피라미드에서도 우수한 성능을 보이며, 특히 낮은 비트율에서는 중요한 계수들을 부호화하는데 비트를 더 할당해 줄 수 있기 때문에 우수한 성능을 보인다.

5. 참고문헌

- [1] J.M Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445-3462, Dec 1993
- [2] A. Said and W.A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuit and Systems for Video Technology*, vol.6, pp.243-250, June 1996
- [3] S. Servetto, K. Ramchandran, and M.T. Orchard, "Image coding based on morphological representation of wavelet data," *IEEE Trans. Image Processing*, vol. 8, pp. 1161-1174, 1999
- [4] B.-B. Chai, J. Vass, and X. Zhuang, "Significance-linked connected component analysis for wavelet image coding," *IEEE Trans. Image Processing*, vol.8, pp.774-784, 1999
- [5] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, vol.1, pp.205-220, Apr. 1992