

의료영상을 위한 중복비트 제거를 이용한 SPIHT 알고리즘의 개선에 관한 연구

박재홍*, 양원석**, 박철우***

춘해보건대학 방사선과*, 동아대학교 의료원 영상의학실**, 동부산대학 전자정보통신과***

A study on improvement of SPIHT algorithm using redundancy bit removing for medical image

Jaehong Park*, Wonseok Yang**, Chulwoo Park***

Dept. Radiological Technology, Choonhae College of Health Science*, Dept. of Radiology, Dong-A University Hospital**,
Dept. of Electronic Information Communication, Dong-Pusan College***

요 약

초기치에서 발생되어지는 비트의 소비를 저주파영역에서의 부호화하는 과정을 없애고, 각 대역별 임계값을 통해, 비트정렬시 LIS에서 발생되어 질 수 있는 비트 소비를 줄일 수 있었다. 그리고 마지막 단계로 허프만 코딩을 적용하여 압축효율을 향상시켰다. 실험 결과를 통해, 특히 낮은 비트율에 대해서 비트율의 향상을 가져오게 되어 전송시에 있어서 더욱 점진적이고 프로그레시브한 전송을 가능하게 되었다.

중심단어 : SPIHT, 허프만 코딩, Wavelet 변환

Abstract

This paper presents improvement of compression rate for SPIHT algorithm based on wavelet compression through redundancy bit removing. Proposed SPIHT algorithm uses a method to select of optimized threshold from feature of wavelet transform coefficients and removes sign bit only if coefficient is LL area. Finally Proposed SPIHT algorithm applies to Huffman coding. Experimental results show that the proposed algorithm achieves more improvement bit rate and more fast progressive transmission with low bit rate.

Key words : SPIHT, Huffman coding, Wavelet transform

I. 서 론

최근 인터넷을 기반으로 하는 디지털 멀티미디어의 급속한 발전으로 인하여, 제안된 대역폭에서의 방대해진 비트 스트림으로 인한 전송속도의 문제와, 고압축

으로 인한 영상정보의 손실 등의 문제가 발생한다. 이러한 방대해진 데이터를 손실을 최소로 하면서 압축 효율을 높이고, 빠른 전송하기 위한 다양한 영상압축 방법들이 나오고 있다. 그중 대표적인 알고리즘으로써, Shapiro가 제안한 EZW(embedded zero tree wavelet)^[1]와 Pearlman이 제안한 SPIHT (set partitioning

hierarchical tree)^[2], 그리고 JPEG2000 표준인 EBCOT(Embedded Block Coding with Optimized Truncation of the embedded bit streams)^[3]등을 들 수 있다.

이러한 웨이블릿 기반 영상부호화 기법은 공간 및 주파수의 해상도를 동시에 표현함으로써 영상의 중요한 특성 중의 하나인 에지부분과 평탄한 부분에 대하여 효과적으로 다룰 수 있으며, 계수의 증가에 따라 계산량도 선형적으로 증가하므로 블록변환 기법을 사용할 필요가 없으며 낮은 비트율에 대하여 DCT기반인 JPEG과 비교 해 볼 때 블록킹(blocking effect) 현상이 제거되고, 상당히 낮은 비트율(low bit rate)에 대해서 우수한 성능을 나타내고 압축을 또한 뛰어나다. 또한 이러한 방법들은 제로트리 구조를 이용하여, 부분 정렬에 의한 중요비트를 먼저 보내는 방법을 이용하므로, 전송시에는 점진적인(progressive) 전송이 가능하고 원하는 비트율에서 정확히 복원할 수 있다.

SPIHT는 EZW를 개선한 방법으로 EZW는 반드시 산술부호화를 필요로 하는 반면 SPIHT는 산술부호기를 사용하지 않고도 보다 우수한 성능을 나타내고 있으며, 또한 EBCOT와 비교해 볼 때 계산적인 복잡도가 작기 때문에 효율적 인코딩 수행이 가능한 장점을 가지고 있다^[4].

본 논문에서는 SPIHT 알고리즘의 구조적인 측면을 분석하여 각각의 단계별로 발생할 수 있는 계수들의 여분의 비트들을 감소시키기 위하여 부모와 자식 그리고 자손들 간의 임계값 설정과 허프만 부호화를 적용하여, 압축효율을 향상시킬 수 있는 방법을 제안한다.

II. 웨이블릿 기반 SPIHT 알고리즘

SPIHT 알고리즘에서의 인코더와 디코더에서 부호화 과정은 크게 4단계 나누어진다. 1)초기화과정(initialization)과 영상에서의 중요한 성분들을 구성하고 있는 부분들을 먼저보내기 위한 2)부분정렬화과정(partial ordering), 정렬된 비트열들을 세분화하기 위한 3)세분화과정(refinement)과 그리고 4)양자화 단계 갱신(quantization-step update)로 구성되어 있다. 표 1은 중요 계수 순으로 정렬된 계수의 리스트를 바이너리 표현

을 나타낸다. 첫 번째 행은 계수의 부호를 나타내고, 나머지 행은 비트평면에 해당하며, 가장 상위행은 MSB(most significant bit)로 중요성이 가장 높은 비트에 해당하고, 가장 하위행 LSB(least significant bit)로 중요성이 가장 적은 비트를 나타낸다. 이렇게 SPIHT 알고리즘은 각 행의 중요비트(화살표)를 먼저 보냄으로써, 전송시 점진적이고, 프로그레시브한 전송이 가능하다.

Table 1. Important factor by transfer coefficient

LIT	Row sign	s	s	s	s	s	s	s	s	s	s	s
MSB	5	1	0	0	0	0	0	0	0	0	0	0
	4	→	1	1	0	0	0	0	0	0	0	0
	3	→	→	1	1	1	0	0	0	0	0	0
	2	→	→	→	→	1	1	1	1	1	1	1
	1	→	→	→	→	→	→	→	→	→	→	→
LSB	0	→	→	→	→	→	→	→	→	→	→	→

그림 1은 2단계 웨이블릿 변환된 영상에서의 부대역간의 공간적 자기유사성을 이용한 공간-방향 계층 트리구조에서의 부모(parent)와 자식(offspring)과의 관계를 나타내고 있다.

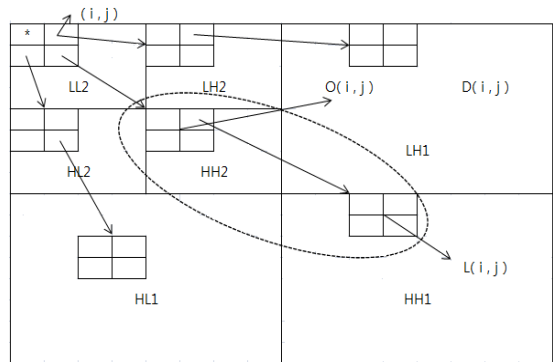


Figure 1. Spatial Orientation Trees in SPIHT

별표 표시 부분을 제외한 LL2영역의 각각의 좌표는 트리의 부모가 되고 다음 레벨에서의 공간적 위치에 있는 4개의 자식을 가지며 항상 2X2 이웃 계수들의 그룹을 형성하게 된다.

그렇게 되면 트리의 수는 LL2 영역의 3/4 크기가 되고 공간적 위치 (i, j)에서의 계수들은 (2i, 2j), (2i+1, 2j), (2i, 2j+1), (2i+1, 2j+1) 위치에서 4개 계수들의 부

모가 된다. [그림 II-2]는 SPIHT 알고리즘의 블록도를 나타낸다. 먼저 초기화 과정은 식(1)을 통해 n 을 구하고 식(2)를 구하여, 임계값(Threshold, T)을 선택하는 것에 의해 시작된다.

$$n = \lceil \log_2 \max_{i,j} (C_{i,j}) \rceil \dots\dots\dots \text{식(1)}$$

$$T = 2^{-n} \dots\dots\dots \text{식(2)}$$

여기서, $C(i, j)$ 는 좌표 (i, j) 에서의 웨이블릿 계수이다. 위에서 구한 임계값을 이용하여 임계값보다 크면 중요계수(significant coefficient) 그리고 작으면 비중요계수(insignificant coefficient)로 구분하게 된다. 그리고 엔코더와 디코더에서 똑같은 방법으로 계수의 중요도를 판별하기 위해 중요한 계수들의 목록인 LSP(list of significant pixels)와 중요하지 않은 계수들의 목록인 LIP(list of insignificant pixels), 그리고 중요하지 않은 집합들의 목록인 LIS(list of insignificant sets)라 불리는 3개의 리스트를 사용한다. 여기서 LIP와 LSP에서의 좌표 (i, j) 는 각각의 계수를 나타내고 LIS는 아래의 조건에 따르는 $D(i, j)$ (type-A)와 $O(i, j)$ (type-B)를 나타낸다.

$$L(i, j) = D(i, j) - O(i, j) \dots\dots\dots \text{식(3)}$$

$O(i, j)$: (i, j) 의 자식(offspring)들의 집합

$D(i, j)$: (i, j) 의 자손(descendants)들의 집합

분류과정은 LIP에 있는 모든 계수들을 조사해 가면서 임계값(T)보다 크면 LSP로 이동시키고, LIS에서도 집합들의 중요도를 판별하여 중요한 집합인 경우 리스트에서 제거되고 4개의 계수들에 대해 중요하면 LSP로 이동시키고 중요하지 않으면 LIP로 이동시킨다. 세분화하는 과정은 LSP에 있는 모든 계수들의 n 번째 MSB(most significant bit)를 차례대로 세분화하여 세분화 정보를 만든다. 마지막으로 양자화 단계 갱신 과정에서는 임계값을 반으로 줄여가면서 원하는 비트율에 도달할 때까지 반복하게 된다.

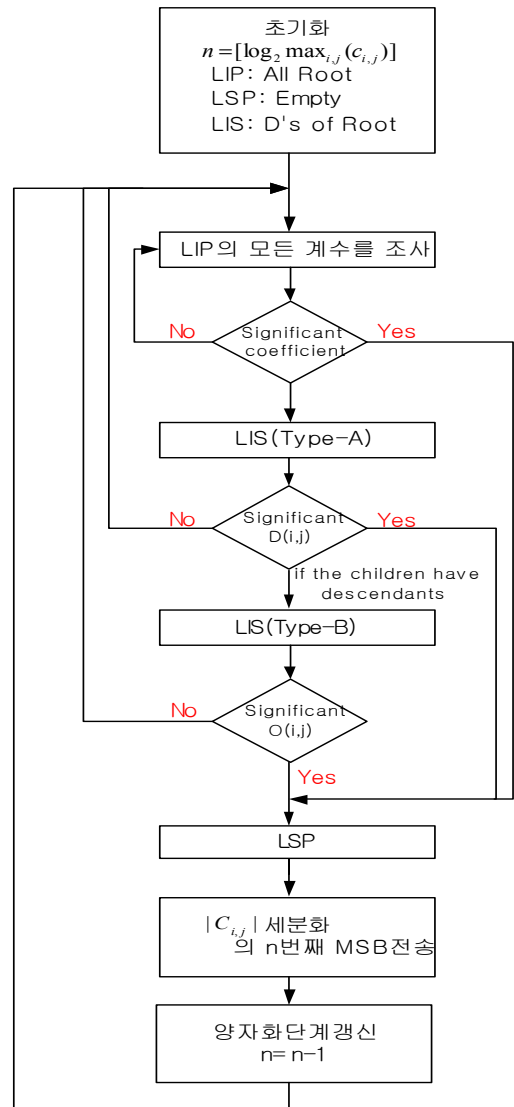


Figure 2. Block diagram of the SPIHT algorithm.

III. 제안한 SPIHT 알고리즘

본 논문에서는 SPIHT 알고리즘의 구조적인 문제점과 웨이블릿 변환된 계수들의 특징을 분석하여 초기치에서의 각 단계에서 발생되어지는 여분의 비트의 소모를 감소시키고, 허프만 코딩을 통해 전체적인 압축효율을 높이고 더 나은 코딩 수행능력을 얻기 위한 방법을 제시한다.

1. 저주파대역에 대한 부호화 제거

기존의 SPIHT 알고리즘은 계수의 중요도를 판별하기 위하여, 1비트를 할당하고, 또한 중요계수에 대하여 양수인지 음수인지를 판별하기 위하여 1비트를 할당하게 된다. 그러나 웨이블릿 변환된 계수들 중 저주파대역은 항상 양수이기 때문에 부호화 할 필요가 없으므로 할당되는 1비트를 감소시킬 수 있었다.

2. LIS(type-A)에서의 임계값 설정

SPIHT 알고리즘에서 고정된 임계값을 가지고 LIP에 있는 목록의 중요도를 판별한 후 모든 자손(descendants)의 목록인 $D(i, j)$ (type-A)를 판별해 나가게 된다. 그러나 여기서 Type-A에서의 임계값과 초기 임계값은 표 1에서처럼 차이가 있으며, 결국은 자손(descendants)들이 중요해질 때까지 중복적으로 트리의 수 만큼 1비트를 낭비하게 된다.

Table 2. Compared to the threshold of each step

T	LENA	Goldhill	Airplane	Hat
초기치	1024	1024	1024	1024
LH	479	380	428	505
HL	380	309	678	581
HH	272	201	320	300

따라서 본 논문에서는 표2에서처럼 다수의 영상에 대한 통계적인 특징들을 살펴 볼때 LH, HL, HH영역의 임계값은 서로 다르다는 사실을 이용하여 초기에 LIS에서의 좌표는 갖지 않고, 그림3에서와 같이 LH영역(LH3, LH2, LH1) HL영역(HL3, HL2, HL1) 그리고 HH영역(HH3, HH2, HH1)에 대하여, 각각의 대표하는 제일 큰 계수값을 구하여 임계값으로 설정 한 후 초기 임계값을 반으로 줄여가면서 자손들의 영역의 대표 임계값(max T)과 비교해 가면서 각각의 영역에 대한 임계값과 일치 할때 LIS에서 자손들의 좌표를 갖게 함으로써 모든 자손들에 대한 적합한 임계치를 가지게 되어 비트를 줄일 수 있었다

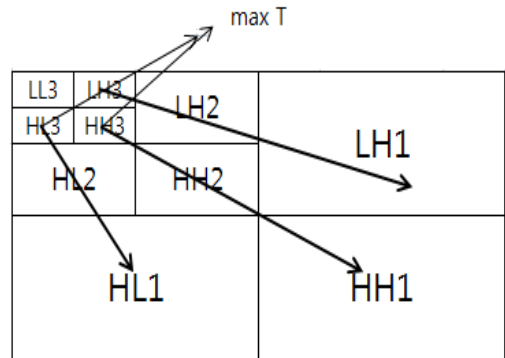


Figure 3. Compared to the threshold (max T) of each step.

3. LIS(Type-B)에서의 임계값 설정

LIS(type-B)에서 집합들이 중요하다고 판단될 때, 그것은 리스트에서 제거되어지고 분할되어진다. 그리고 4개의 새로운 계수들은 현재의 임계값에 따라 판별되어지고 LIP나 LSP로 이동하게 된다.

이러한 과정에 있어서 이 새로운 계수들이 분할되어야 할 임계값과 현재의 임계값은 표 3에서 처럼 웨이블릿 변환된 영상들의 특징을 살펴 볼때 같거나, T/2 혹은 T/4보다 작기도 하다. 만일 T/2보다 작다면 현재의 임계값에서 처리 않고 다음단계의 임계값에서 처리 할 수 있으므로 1비트를 절약 할 수 있다.

Table 3. Compared between the offspring of the threshold

T	LENA	Airplane	Hat	Camera
descendants	256	512	512	512
offspring(1)	256	256	256	256
offspring(2)	128	128	64	128

따라서 본 논문에서는 그림 4와 같이 각각의 영역 별간의 대표 임계값, 즉 max_T(1)과 max_T(2)를 설정하여 T/2 혹은 T/4보다 작은 경우에 대하여 발생할 수 있는 비트의 소모를 줄일 수 있었다.

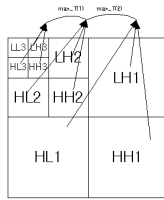


Figure 4. Compared to the threshold (max_T(1),max_T(2)) of each step.

마지막 단계로, 초기 양자화 과정에서부터 6단계까지는 전체적인 비트가 중요하지 않기 때문에 연속되는 0이 많이 발생하게 된다. 이러한 사실을 통해 허프만 부호화[9]를 적용하여 비트수를 감소시켰다.

IV. 실험 및 고찰

본 논문에서 제안한 압축방법과 기존의 SPIHT 알고리즘의 압축효율을 비교하기 위하여 256X256 그레이 레벨을 갖는 영상을 이용하였고 3단계 웨이블릿 변환한 SPIHT 알고리즘을 이용하여 평가하였다.

$$bit\ save = \frac{\text{proposed SPIHT bit rate}}{\text{original SPIHT bit rate}} \times 100$$

Table 4. original SPIHT(O_SPIHT) and proposed SPIHT(P_SPIHT)의 bit rate



(a)원 영상



(b)wavelet
PSNR=50dB



(c)n=10
O_SPIHT:244263bit
P_SPIHT:237539bit



(d)n=9
O_SPIHT:169231bit
P_SPIHT:162502bit



(e)n=8
O_SPIHT:103696bit
P_SPIHT:96962bit



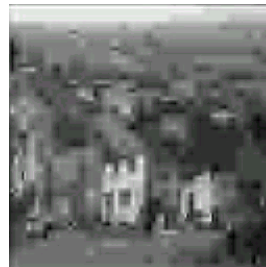
(f)n=7
O_SPIHT:56643bit
P_SPIHT:49904bit



(g)n=6
O_SPIHT:28808bit
P_SPIHT:22420bit



(h)n=5
O_SPIHT:15230bit
P_SPIHT:9591bit



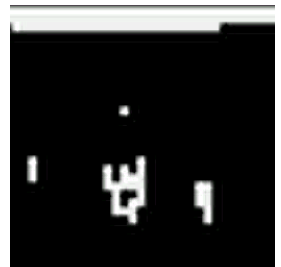
(i)n=4
O_SPIHT:9746bit
P_SPIHT:4781bit



(j)n=3
O_SPIHT:7129bit
P_SPIHT:2607bit



(k)n=2
O_SPIHT:4848bit
P_SPIHT:1550bit



(l)n=1
O_SPIHT:2171bit
P_SPIHT:607bit

표 4는 원 SPIHT와 제안한 방법의 SPIHT 알고리즘의 이용한 비트율을 나타내고 있으며, 2.76%의 비트율 향상을 가져왔다

Table 5. Applied to each bit save

	(1)방법	(2)방법	(3)방법	허프만 적용시
LENA	1024bit	2329bit	536bit	3203bit
HAT	1020bit	2350bit	572bit	3085bit
GoldHill	1014bit	2704bit	404bit	3245bit
Airplane	1024bit	2003bit	464bit	2413bit

Table 6. Original image and several images of the experimental results

step	구 분	Lena	Airplane	Hat
n=1	O_SPIHT	2145	2822	2428
	P_SPIHT	628	991	832
n=2	O_SPIHT	4703	5116	4881
	P_SPIHT	1593	1849	1679
n=3	O_SPIHT	7184	7313	7256
	P_SPIHT	3086	3165	3207
n=4	O_SPIHT	10430	10841	10835
	P_SPIHT	5690	6107	6226
n=5	O_SPIHT	17725	19548	18463
	P_SPIHT	12296	14042	13083
n=6	O_SPIHT	32656	36983	33930
	P_SPIHT	26418	31057	27482
n=7	O_SPIHT	57590	64766	61389
	P_SPIHT	51191	58845	54107
n=8	O_SPIHT	90100	102588	104135
	P_SPIHT	87281	96672	96853
n=9	O_SPIHT	142445	153537	163731
	P_SPIHT	139626	147626	156450
n=10	O_SPIHT	219564	220611	235810
	P_SPIHT	213180	214705	228531
bit save		2.91%	2.59%	3.09%

V. 결 론

본 논문에서는 초기치에서 발생되어지는 비트의 소비를 저주파영역에서의 부호화하는 과정을 없애고, 각 대역별 임계값을 통해, 비트정렬시 LIS에서 발생되어질 수 있는 비트 소비를 줄일 수 있었다. 그리고 마지막 단계로 허프만 코딩을 적용하여 압축효율을 향상시켰다. 실험 결과를 통해, 특히 낮은 비트율에 대해서 비트율의 향상을 가져오게 되어 전송시에 있어서 더욱 점진적이고 프로그레시브한 전송을 가능하게 되었다.

감사의글

본 연구는 2011년도 춘해보건대학 학술연구비 지원에 의한 것입니다.

참 고 문 헌

- [1] A. Said and W. A. Pearlman, "A new fast/efficient image codec based on set partitioning in hierarchical trees," IEEE Trans. Circuit System Video Technol, vol. 6, pp.243-250, 1996.
- [2] J. M. shapiro, "Embeded image coding using zerotrees of wavelet coefficient", IEEE Transactions on Signal Processing, Vol. 41, No.12, pp.3445-3262, 1993.
- [3] David Taubman, "High Performance Scalable Image Compression with EBCOT", IEEE Transaction on Image Processing, Vol 6, pp. 243-250, 1996.
- [4] 4. Ulug Bayazit and William A. Pearlman , "Algorithmic modifications to SPIHT", IEEE Trans. Image Processing, Vol.3, pp.800-803, 2001.
- [5] Jian Zhu and Lawson, "Improvements of the SPIHT for image code by wavelet transform" IEEE Trans. pp.24/1-24/5, 2000.
- [6] David Salomon, Data Compression The Complete Reference Second Edition, 2001.
- [7] Amir Said, Example of Application for Image Compression, 1999,
- [8] Z.Xiong, K. Ramchandran, and M, T. Orchard, "Space-Frequency Quantization for wavelet image coding", IEEE Trans. on Image Proc, Vol. 6, No.5, pp 677-693, 1997.