프로그레시브 영상 전송에 적합한 SPIHT알고리즘의 개선에 관한 연구

설 경 호, 조 상 현, 주 동 현, 김 태 형, 김 두 영
동아대학교 전자공학과 영상실험실

A study on improvement of SPIHT algorithm for progressive image transmission

Kyung-Ho Seol, Sang-Hyun Jo, Dong-Hyun Joo, Tae-Hyung Kim, Doo-Young Kim
DongA University
E-mail : skhtop@hanmail.net

요 약

본 논문에서는 웨이블렛 압축 기반인 SPIHT 알고리즘의 압축효율을 개선시킬 수 있는 방법을 제안한다. 특히 초기치에서 발생될 수 있는 비트소비를 줄임으로써, 낮은 비트율에서 더욱 빠른 점진적인 전송을 구현하였다. 트리구조에 따르는 최적화된 임계값 설정과 웨이블렛 변환된 계수들의 계수값을 통해 비트율 향상을 얻을 수 있었다.

I. 서 론

최근에는 웨이블렛변환을 이용한 다양한 영상압축 방법들이 나오고 있으며, 그 중 대표적인 알고리즘으로써, Shapiro가 제안한 EZW(embedded zero tree wavelet)[2]와 Pearlman이 제안한 SPIHT(set partitioning in hierarc-hical tree)[1], 그리고 JPEG2000표준인 EBCOT(Embed-ded Block Coding with Optimized Truncation of the embedded bit-streams)[3]등을 들 수 있다.

이러한 웨이블렛 기반 부호화 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 현상이 제거되고, 상당히 낮은 비트율(low bit rate)에 대해서 우수한 성능을 나타내고 압축율 또한 뛰어나다.

또한 웨이블렛 압축기반은 zero tree 구조를 이용하여, 부분정렬에 의한 중요비트를 먼저 보내는 방법을 이용하므로, 전송시에는 점진적인(progressive) 전송이 가능하고 원하는 비트율에서 정확히 복원 할 수 있다.

SPIHT는 EZW를 개선한 방법으로 EZW는 반드시 산술부호화를 필요로 하는반면 SPIHT는 산술부호기를 사용하지 않고도 보다 우수한 성능을 나타내고 있으며, 또한 EBCOT와 비교 해 볼 때 계산적인 복잡도가 작기 때문에 효율적인코딩 수행이 가능한 장점을 가지고 있다.[4]

본 논문에서는 SPIHT 알고리즘의 구조적인 측면을 분석하여 각각의 단계별로 발생할 수 있는 계수들의 여분의 비트들을 감소시킴으로써 압축효율을 향상시킬 수 있는 방법을 제안한다. 본 논문의 구성은 II장 에서는 SPIHT방식의 압축알고리즘에 관해서 기술하고, III장 에서는 제안한 SPIHT 알고리즘을 기술하며, IV장 에서는 실험 및결과를 살펴보고 V장 에서는 결론을 맺는다.

II. 웨이블렛기반 SPIHT 알고리즘

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

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.

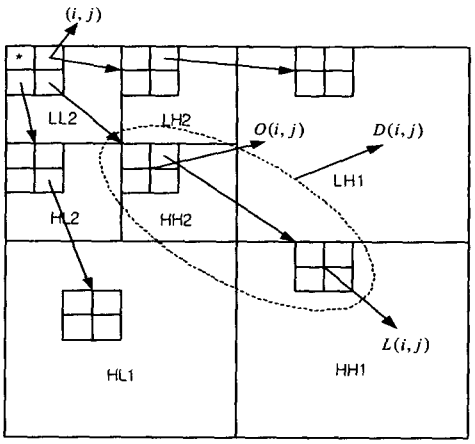


그림 1. Spatial Orientation Trees in SPIHT

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 부모가 된다.

부호화 하는 과정은 초기화 과정(initialization)과 분류과정(sorting pass), 세분화과정(refinement), 그리고 양자화단계 갱신(quantization-step update)등 4개의 단계로 나누어진다.

그림 2는 SPIHT 알고리즘의 블록도를 나타낸다.

먼저 초기화 과정은 아래의 식을 통해 n을 구하고 임계값(Threshold, T)을 선택하는 것에 의해 시작된다.

$$n=[\log_2 \max_{i,j}(C_{i,j})] \quad (1)$$

$$T=2^n \quad (2)$$

여기서 $C(i,j)$ 는 좌표 (i,j) 에서의 웨이블렛 계수이다. 위에서 구한 임계값을 이용하여 임계값보다 크면 중요계수(significant coefficient) 그리고 작으면 비 중요계수(insignificant coefficient)로 구분하게 된다.

그리고 엔코더와 디코더에서 똑같은 방법으로 집합의 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.

집합의 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.

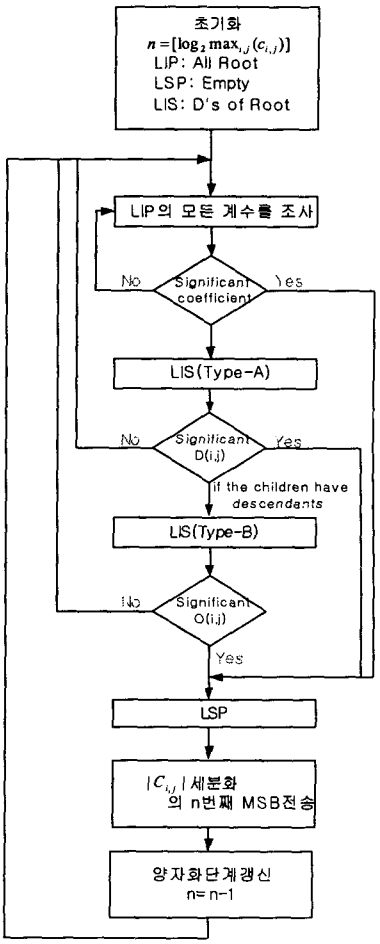


그림 2. SPIHT 알고리즘의 블록도

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.

$$L(i,j)=D(i,j)-O(i,j) \quad (3)$$

$O(i,j)$: (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 n번째 MSB(most significant bit)를 차례대로 세분화하여 세분화 정보를 만든다. 마지막으로 양자화 단계 갱신 과정에서는 임계값을 반으로 줄여가면서 원하는 비트율에 도달할때 까지 반복하게 된다.

III. 제안한 SPIHT 알고리즘

본 논문에서는 SPIHT 알고리즘과 웨이블렛 변환된 계수들의 특징을 분석하여 초기치에서의 각 단계에서 발생되어지는 여분의 비트를 줄임으로써, 압축효율을 높이고 더 나은 코딩 수행능력을 얻기 위한 방법을 제시한다.

가. LL영역의 부호비트제거

SPIHT 알고리즘에서 LIP에서는 모든 계수들은 중요도를 판별하기 위해 1비트를 할당하게 되고 또한 만일 중요한 계수 값이라면 양수(positive)인지 음수(negative)인지를 판별하기 위하여 1bit를 할당하게 된다.

그러나 초기치에 LIP에는 부모(parent)에 해당하는 LL영역의 계수들의 좌표만 가지고 있다. 여기서 웨이블렛 변환된 계수들중 LL영역에서는 항상 양수라는 것을 예측할 수 있다.

LL영역을 처리하기위한 LIP와 LIS에서 발생되어진 중요하지 않은 계수에 대한 처리를 위한 LIP를 두어서 처리한다면 LL영역의 크기만큼 부호비트를 줄일 수 있다.

나. LIS(type-A)에서의 최적화된 임계치

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

따라서 본 논문에서는 표 1에서처럼 다수의 영상에 대한 통계적인 특징들을 살펴 볼때 LH, HL, HH영역의 임계값은 서로 다르다라는 사실을 이용하여 초기에 LIS에서의 좌표는 갖지 않고, LL영역을 제외한 LH영역 HL영역 그리고 HH영역에 대하여, 각각의 대표하는 제일 큰 계수값을 구하여 임계값으로 설정 한 후 초기 임계값을 반으로 줄여가면서 자손들의 영역의 임계값과 비교해 가면서 각각의 영역에 대한 임계값과 일치 할때 LIS에서 자손들의 좌표를 갖게 함으로써 모든 자손들에 대한 적합한 임계치를 가지게 되어 비트를 줄일 수 있었다.

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

표 1. 각 대역별 임계치(Threshold)비교

다. LIS(Type-B)에서의 최적화된 임계치

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

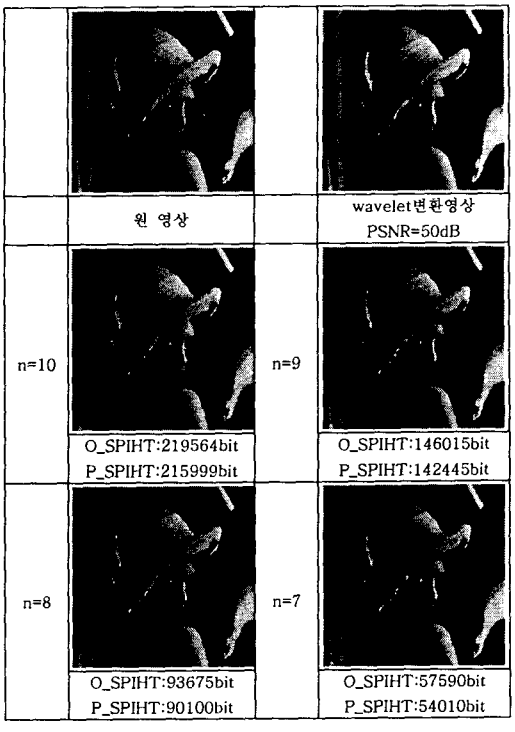
이러한 과정에 있어서 이 새로운 계수들이 분할되어져야 할 임계값과 현재의 임계값은 표 2에서처럼 웨이블렛 변환된 영상들의 특징을 살펴 볼때 같거나, T/2 혹은 T/4보다 작기도 하다. 만일 T/2보다 작다면 현재의 임계값에서 처리 않고 다음 임계값에서 처리 할 수 있으므로 1bit를 절약 할 수 있다.

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

표 2. 자손들 간의 임계치 비교

IV. 실험 및 고찰

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



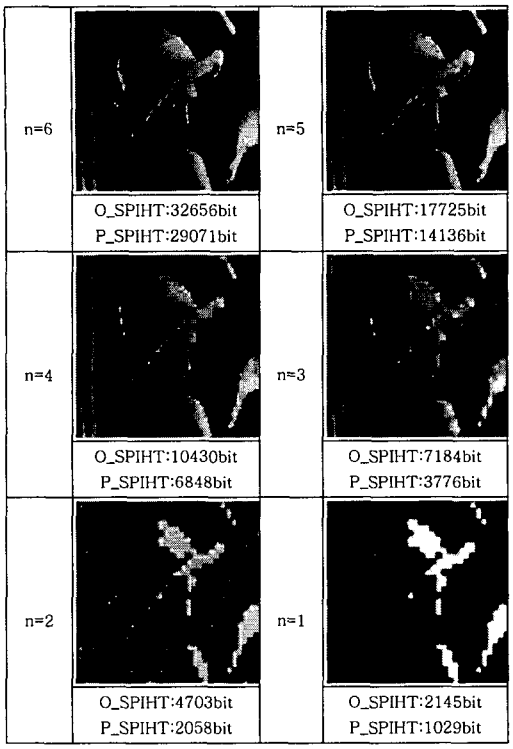


표 3. 원 SPIHT(O_SPIHT)와 제안한 SPIHT (P_SPIHT)의 bit rate

표 3은 원 SPIHT와 제안한 방법의 SPIHT 알고리즘의 이용한 bit rate을 나타내고 있으며, 1.63%의 bit율 향상을 가져왔다.

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

step		Camera	Airplane	Hat
n=1	O_SPIHT	2317	2822	2428
	P_SPIHT	1029	1029	1029
n=2	O_SPIHT	4870	5116	4881
	P_SPIHT	2058	2379	2058
n=3	O_SPIHT	7272	7313	7256
	P_SPIHT	4045	4059	3794
n=4	O_SPIHT	11303	10841	10835
	P_SPIHT	8056	7605	7257
n=5	O_SPIHT	19445	19548	18463
	P_SPIHT	16087	16316	14873
n=6	O_SPIHT	34538	36983	33930
	P_SPIHT	31167	33756	30323
n=7	O_SPIHT	60403	64766	61389
	P_SPIHT	57032	61544	57780
n=8	O_SPIHT	95401	102588	104135
	P_SPIHT	92034	99371	100526
n=9	O_SPIHT	139621	153537	163731
	P_SPIHT	136258	150325	160123
n=10	O_SPIHT	204366	220611	235810
	P_SPIHT	201008	217404	232204
bit save		1.65%	1.46%	1.53%

표 4. 원 영상과 여러 가지 영상의 실험 결과

표 4는 원 SPIHT와 다른 영상들 간의 비트율을 실험한 결과이다.

V. 결 론

본 논문에서는 SPIHT 알고리즘에서 중복적인 비트제거를 통해 압축효율을 높일 수 있었다.

특히 초기치에서 발생되어지는 비트의 소비를 LL영역에 대해서는 부호화하는 과정을 없애고, 각 대역별 임계값을 통해, 비트정렬시 LIS에서 발생되어 질 수 있는 비트 소비를 줄일 수 있었다.

실험 결과를 통해, 특히 낮은 비트율에 대해서 비트율의 향상을 가져오게 되어 전송시에 있어서 더욱 점진적인 전송을 가능하게 되었다.

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. Video Technol.vol. 6, pp.243-250,1996.

[2] j. M. shapiro, "Embaded image coding using zerotrees of wavelet coefficient", IEEE Transactions on Signal Processing, Vol. 41, NO.12, pp.3445 3262 Dec. 1993.

[3] David Taubman, "High Performance Scalable Image Compression with EBCOT", IEEE Transaction on Image Processing, Vol 6, pp. 243-250, 1966.

[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"

[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, May 1997.