

SPIHT 영상코딩 알고리즘의 시간복잡도 해석

Time Complexity Analysis of SPIHT(Set Partitioning in Hierarchy Trees) Image Coding Algorithm

박 영 석

경남대학교 정보통신공학부

Young-Seak Park

Division of Information and Communication Engineering

Kyungnam University

e-mail: yspark@kyungnam.ac.kr

요 약

EZW(Embedded Zerotree Wavelet) 알고리즘이 소개된 이래 일련의 내장형 웨이브렛 코딩 방법들이 제안되어져왔다. 이들의 하나의 공통된 특징은 EZW 알고리즘의 기본 아이디어를 근간으로 한다는 점이다. 특히 SPIHT(Set Partitioning in Hierarchy Trees) 알고리즘은 이들 중의 하나로서 산술 코더를 사용하지 않더라도 EZW 와 같거나 혹은 더 나은 성능을 제공할 수 있기 때문에 널리 알려져 왔다. 이러한 SPIHT 알고리즘은 계산적으로 단순하면서도 뛰어난 수치적, 시각적 결과를 제공한다. 그러나 SPIHT 알고리즘의 처리 시간복잡도(time complexity)에 대한 평가는 상대적인 비교를 통해 효과적임을 실험적으로 확인한 것에 불과하고 지금까지 엄밀한 시간복잡도 해석이 행하여지지 않았다. 본 연구에서는 SPIHT 알고리즘의 처리 시간복잡도를 해석하고 한 비트평면 처리에 대한 최악의 경우(worst case) 시간복잡도가 $O(n \log_2 n)$ 임을 증명한다.

ABSTRACT

A number of embedded wavelet image coding methods have been proposed since the introduction of EZW(Embedded Zerotree Wavelet) algorithm. A common characteristic of these methods is that they use fundamental ideas found in the EZW algorithm. Especially, one of these methods is the SPIHT(Set Partitioning in Hierarchy Trees) algorithm, which became very popular since it was able to achieve equal or better performance than EZW without having to use an arithmetic encoder. The SPIHT algorithm is computationally very simple, but even so it provides excellent numerical and visual results. But the evaluation of its time complexity is no more than the relative result of experimental comparisons and the strict time complexity analysis wasn't taken until now. In this paper, we analyze strictly the processing time complexity of SPIHT algorithm and prove that the time complexity for one bit-plane processing is $O(n \log_2 n)$ in worst case.

Key words : 내장형 영상코딩, EZW 코딩 알고리즘, SPIHT 알고리즘, SOT 계층구조, SOT 코딩 트리, 시그니피컨스 맵, 시간복잡도

I. 서론

영상압축기법 특히 비가역적(nonreversible) 혹은 손실(lossy) 압축기법들은 하나의 정상 소오스(stationary source)에 대한 무한한 계산량에 따라 최적에 접근한다는

정보이론의 소오스 코딩이론의 관점에 따라 그들이 더욱 효과적인 만큼 계산적으로 더욱 복잡하게 되는 것으로 알려져왔다[1,2].

그럼에도 불구하고 Shapiro[3]에 의해 소개된 내장형 제로트리 웨이브렛(EZW: embedded zerotree wavelet)이라 불리는 영상코딩기법은 효율성과 복잡도(complexity)

의 병행성 논의를 중단시켰다. 이 기법은 가장 복잡한 기법들과 성능에 있어서 경쟁적일 뿐만 아니라 하나의 내장형 비트 스트림(embedded bit stream)을 생성하는데 극단적으로 빠른 실행시간을 가진다. 하나의 내장형 비트 스트림을 가지고 코드 비트들의 수신은 임의의 시점에서 중단될 수 있고 디코딩되어 복원될 수 있다. 그 의미 깊은 업적에 이어 보다 효과적인 EZW기법 기반 원리를 사용하는 웨이브렛 코딩 알고리즘들이 개발되어 왔다 [4-7].

대표적인 것으로 SPIHT(Set Partitioning in Hierarchy Trees)알고리즘[4], EBCOT(Embedded Block Coding with Optimized Truncation of the embedded bit-streams)알고리즘[5], 그리고 SRC(Stack Run Coding) 알고리즘[7]을 들 수 있다. 특히 SPIHT 알고리즘은 그것이 산술엔코더(arithmetic encoder)의 사용 없이도 EZW 보다 같거나 더 나은 성능을 가졌기 때문에 매우 유명하게 되었고 JPEG 2000 표준 결정에 있어서 EBCOT에 대항하는 가장 강력한 경쟁자였다[1]. 2D(2-dimensional)에 이어 그 후 EZW 계열 알고리즘은 3D(3-dimensional)로 확장되고[2,8], 그중에서 3D SPIHT[8]는 어떤 모션보상(motion compensation)없이 계산적으로 단순하고 보다 효과적인 비디오 코딩 시스템의 가능성과 뛰어난 수치적, 시각적 결과를 보임으로써 그 유용성과 발전 가능성을 확대하고 있다.

그러나 SPIHT 알고리즘의 처리 시간복잡도(time complexity)[9]에 대한 평가는 상대적인 비교를 통해 효과적임을 실험적으로 확인한 것에 불과하고 엄밀한 시간 복잡도 해석이 행하여지지 않았다. SPIHT 알고리즘의 중요성에 비추어 컴퓨팅이론 관점의 시간복잡도 해석은 선행 혹은 후속 연구의 비교 분석에 있어서 파급효과가 매우 클 것으로 예상된다. 따라서 본연구에서는 SPIHT 알고리즘의 이러한 발전 가능성과 파급효과에 주목하고 SPIHT 알고리즘의 시간복잡도를 분석하고 최악의 경우 시간복잡도를 유도하고자 한다.

II. SPIHT 알고리즘 분석

1. SOT 계층구조

(Spatial Orientation Tree Hierarchy)

SPIHT 알고리즘은 프로그래시브 전송 알고리즘과 EZW 코딩 개념에 기반을 둔 웨이브렛 변환 계수 이미지에 대한 비트 평면의 코딩 알고리즘이다[4]. 프로그래시브 전송 알고리즘은 크게 대별해서 소팅패스(sorting pass), 리파인먼트 패스(refinement pass)로 이루어지며 SPIHT 알고리즘은 개선된 소팅패스에 해당한다.

SPIHT 알고리즘은 그림 1과 같은 웨이브렛 변환 이미지에 대한 SOT(Spatial Orientation Tree)[2] 즉, 시그니피칸스-맵(significance map)의 계층구조의 개념을 사용하여 이미지의 각 비트평면(bit plane)에 반복적으로 적용되어 변환 이미지를 효과적으로 압축한다. 이후 논의의

간결성을 위하여 하나의 비트평면, 즉 시그니피칸스 맵의 코딩에 논의를 집중한다. 그림 2는 3-레벨 웨이브렛 변환 이미지의 한 예[4]이고 그림3은 그림2의 변환 이미지에 문턱치 32를 적용한 시그니피칸스-맵 혹은 MSB(Most Significant Bit) 비트평면을 보인다. 설명을 위해 그림 1의 계층적 SOT 트리 구조에 대한 다음의 정의를 사용한다. 이것은 Said and Pearlman[3]의 정의를 확장한 것이다.

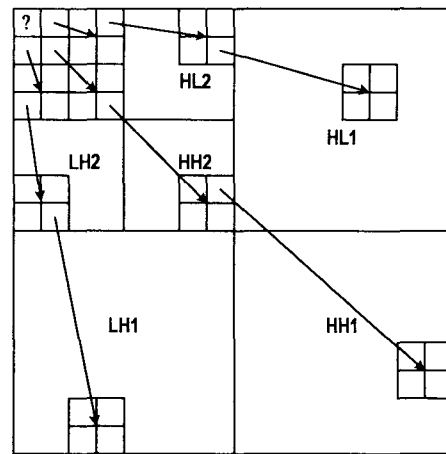


Fig. 1. The hierarchical SOT structure of significance map[3]

$O(\{(i,j)\})$: 노드 (i,j) 집합의 모든 직계 일세대순(offspring) 좌표의 집합이며, $\{(i,j)\}$ 가 단일 좌표집합이면 $O(i,j)$ 로 표기한다.

$D(\{(i,j)\})$: 노드 (i,j) 집합의 모든 자손(descendants) 좌표의 집합이며, $\{(i,j)\}$ 가 단일 좌표집합이면 $D(i,j)$ 로 표기한다.

$L(\{(i,j)\})$: $D(\{(i,j)\}) - O(\{(i,j)\})$ 이며, $\{(i,j)\}$ 가 단일 좌표집합이면, $L(i,j)$ 로 표기한다.

$P(\{(i,j)\})$: 노드 (i,j) 집합의 부모(parent) 좌표의 집합이며, $\{(i,j)\}$ 가 단일 좌표집합이면 $P(i,j)$ 로 표기한다.

그리고 임의의 좌표 집합 즉, 파티션(partition) T 의 시그니피칸스를 나타내기 위해 식 (2)의 함수 $S_n(T)$ 를 사용한다. 즉, $S_n(T)$ 는 파티션 T 의 계수 중 그 절대값이 문턱치 2^n 보다 클 때 1을 그렇지 않으면 0의 값을 가진다. 단일 화소 집합의 표현을 단순화하기 위해 $S_n(\{(i,j)\})$ 을 $S_n(i,j)$ 로 표기한다.

$$S_n(T) = \begin{cases} 1, & \max_{(i,j) \in T} \{|C_{ij}|\} \geq 2^n \\ 0, & \text{Otherwise} \end{cases} \quad (1)$$

2. SOT 코딩트리(Coding Tree) 구조

SPIHT 알고리즘은 엔코더(encoder)의 코딩 실행경로(execution path)를 디코더에서 재현함으로써 전송되는

시그니피컨스-맵의 코딩정보를 통해 시그니피컨스-맵을 재구성한다.[4] 이것은 엔코더에서 생성되어 전송되는 데이터의 구조분석을 통해 보다 분명하고 쉽게 이해할 수 있다. 결과적으로 SPIHT 알고리즘은 시그니피컨스-맵에 대한 코딩으로 소팅 패스에서 그림 4와 같은 데이터 구조 정보 즉, SOT 코딩 트리(SOT coding tree)를 생성하고 부호비트(sign bit)와 함께 전송한다. 여기서 루트 노드(root node)의 정보가 3비트인 것은(원 알고리즘에서는 1비트) 원 알고리즘의 완전성을 위해 수정 표현되었다. 이 SOT 코딩트리는 SOT 코딩트리 상의 노드에 대해 다음과 같은 특징을 가진다.

- 1) 노드 (i,j) 는 $S(i,j)=1$ 혹은 $S(D(i,j))=1$ 혹은 $O(P(i,j))=1$ 이다.
- 2) 노드 (i,j) 가 $L(P(i,j)) \neq \emptyset$ 그리고 $S(L(P(i,j)))=0$, 혹은 $L(P(i,j)) \in \emptyset$ 그리고 $S(O(P(i,j)))=1$ 이면 자신의 시그니피컨스, $S(i,j)$ 를 나타내는 1비트 정보를 가진다.
- 3) 노드 (i,j) 의 $L(P(i,j)) \in \emptyset$ 그리고 $S(O(i,j))=1$ 이면, 노드 (i,j) 는 2비트 b_1b_2 정보를 가지며 b_1 은 $S(i,j)$ 를 그리고 b_2 는 $S(O(i,j))$ 를 나타낸다.
- 4) 2)와 3)의 경우가 아니면, 노드 (i,j) 는 3비트 $b_1b_2b_3$ 정보를 가지며 b_1 은 $S(i,j)$ 를, b_2 는 $S(O(i,j))$ 를, 그리고 b_3 는 $S(L(i,j))$ 를 나타낸다.
- 5) 노드 (i,j) 는 $S(D(i,j))=1$ 일 때에 한해서 루트노드 경우는 3개 그리고 그렇지 않은 경우 4개의 자식(children) 노드를 가진다.

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4

Fig. 2. An example of 3-level wavelet transform[4]

III. SPIHT 알고리즘의 시간복잡도 해석

본 연구의 이미지 크기는 다음의 가정에 따르며 이것은 대부분 영상처리에 있어서 일반적인 가정이다.

[가정] 이미지는 크기 $N \times N = n$ 을 가정하며 여기서 $N=2^i$ 인 양의 정수 i 가 존재한다.

그림 1과 같이 정의되는 계층적 SOT 트리구조의 최대 레벨 수는 l 은 N 을 연속적으로 이분하여 1이 될 때까지의 이등분 횟수에 1을 더한 것으로 정의됨으로 증명 없이 다음의 보조정리를 사용한다.

1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Fig. 3. The significance map for threshold value, 32(MSB Plane)

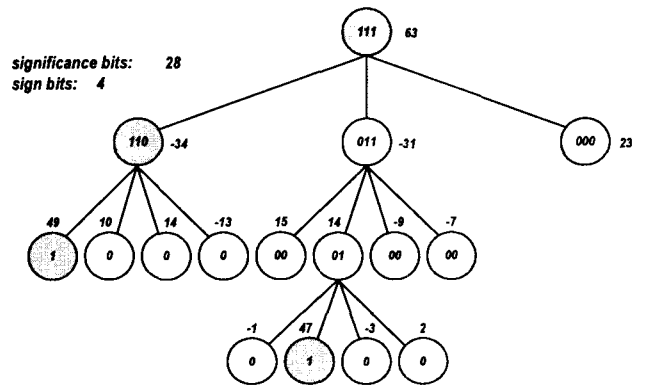


Fig. 4. The SOT coding tree structure for Fig. 3

[보조 정리] 이미지 크기 $N \times N = n$ 인 계층적 SOT 트리구조의 최대 레벨 수 l 은 $\frac{1}{2}(\log_2 n) + 1$ 이다.

SPIHT 알고리즘에서 하나의 비트 평면에 대한 SOT 코딩 트리를 완성하기 위해서는 SOT 계층구조의 각 노드에서 자신의 시그니피컨스, $S(i,j)$ 와 자손의 시그니피컨스, $S(D(i,j))(S(L(i,j))$ 를 포함)를 결정해야하고 최악의 경우 자신을 포함한 자손의 모든 노드를 탐색하여야 한다. 따라서 SPIHT 알고리즘의 최악의 경우 시간복잡도는 최악의 경우 SOT 코딩 트리를 완성하기 위한 노드 탐색의 총 수, N_{∞} 에 의존한다. 그림 5는 최악의 경우 SOT 계층구조 형태를 보이며 하나의 루트와 레벨 1에서 레벨 $l-1$ 에 이르는 3개의 완전 서브쿼드트리(complete subquadtrees) a,b,c로 이루어진다. 그리고 그림 6은 최악의 경우 시그니피컨스

맵의 한 예를 보인다. 여기서 완전 서브콰드트리란 SOT상 하위의 모든 화소(혹은 노드)를 포함함을 의미하며 최악의 경우 SOT 코딩트리의 총 노드 수는 시그니피컨스 맵의 총 화소 수와 같다. SOT 코딩트리의 각 레벨에서 하위 레벨의 모든 노드를 탐색하는 경우를 완전탐색(complete search)라 정의하면 레벨 i 에서 완전탐색의 총 노드 수, N_i 는 다음처럼 유도된다.

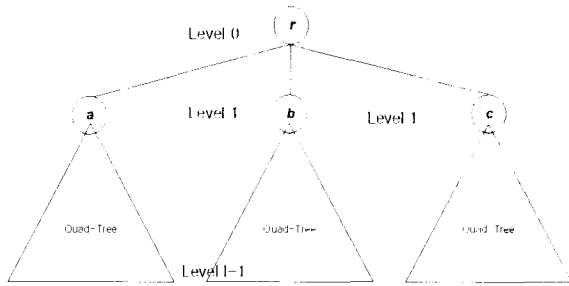


Fig. 5. SOT coding tree structure with 3 subquadrees

0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	0
1	0	0	1	1	0	1	0
0	0	0	0	0	0	0	0
0	1	1	0	0	1	0	1
0	0	0	0	0	0	0	0

Fig. 6. An Example of worst case significance-map

레벨 0에서, N_0 는 레벨 1에서 레벨 $l-1$ 에 이르는 3개의 완전 서브콰드트리의 총노드 수에 루트노드 하나를 더한 값이 된다. 그리고 레벨 $1 \leq i \leq l-1$ 에서, N_i 의 일반식은 귀납적으로 식(2)와 같이 유도된다.

$$\begin{aligned}
 N_0 &= 3(4^0 + 4^1 + \dots + 4^{l-2}) + 4^0 = 4^{l-1} \\
 N_1 &= 3(4^0 + 4^1 + \dots + 4^{l-2}) = 4^{l-1} - 4^0 \\
 N_2 &= 3(4^1 + 4^2 + \dots + 4^{l-2}) \\
 &= 4^1(3(4^0 + 4^1 + \dots + 4^{l-3})) \\
 &= 4^{l-1} - 4^1 \\
 &\vdots \\
 N_i &= 3(4^{i-1} + 4^i + \dots + 4^{l-2}) \\
 &= 4^{i-1}(3(4^0 + 4^1 + \dots + 4^{l-i-1})) \quad (2) \\
 &= 4^{l-1} - 4^{i-1}
 \end{aligned}$$

$$\begin{aligned}
 &\vdots \\
 N_{l-2} &= 3(4^{l-3} + 4^{l-2}) \\
 &= 4^{l-3}(3(4^0 + 4^1)) \\
 &= 4^{l-1} - 4^{l-3} \\
 N_{l-1} &= 3(4^{l-2}) \\
 &= 4^{l-2}(3(4^0)) \\
 &= 4^{l-1} - 4^{l-2}
 \end{aligned}$$

따라서, SOT 코딩트리의 모든 레벨에서 완전탐색을 가정하면 총 탐색 노드 수 N_t 는 각 레벨에서 완전탐색해야 할 총 노드 수들의 총합으로 식(3)과 같다. 이 식을 이용해서 최악의 경우 시간복잡도는 [정리2]로 정리된다.

$$\begin{aligned}
 N_t &= \sum_{i=0}^{l-1} N_i \\
 &= (l)4^{l-1} - (4^0 + 4^1 + \dots + 4^{l-2}) \quad (3) \\
 &= (l)4^{l-1} - \frac{1}{3}(4^{l-1} - 1) \\
 &= (l - \frac{1}{3})(4^{l-1}) + \frac{1}{3}
 \end{aligned}$$

[정리 2] 크기 $N \times N = n$ 인 하나의 비트 평면을 처리하기 위한 SPIHT 알고리즘의 최악의 경우 시간복잡도는 $O(n \log_2 n)$ 이다.

[증명] 변환 이미지의 총화소 수는 $n = N_0$ 이고 최악의 경우 SOT 코딩트리에서 콰드트리의 성질로부터 레벨 $l-1$ 에서의 총 노드 수는 4^{l-2} 이다. 최악의 경우, 각 레벨의 노드 (i, j) 로부터 $S(D(i, j))(S(L(i, j)))$ 의 결정을 포함/결정을 위한 탐색에 있어서 $l-1$ 레벨의 임의의 하나의 노드가 시그니피컨트하면 $S(D(i, j)) = 1$ (혹은 $S(L(i, j)) = 1$)로 결정됨으로 더 이상의 탐색은 필요없다. 그리고 최악의 경우라면 $l-1$ 레벨에는 시그니피컨트 노드가 반드시 존재한다. 따라서 최악의 경우 탐색 노드의 총수, N_w 는 다음의 식(4)를 만족한다.

$$N_t - l(3(4^{l-2})) < N_w < N_t \quad (4)$$

보조정리로부터 SOT 트리구조의 최대 레벨 수는 $l = \frac{1}{2}(\log_2 n) + 1$ 이다. 그리고 전술한 식(3)으로부터 N_t 는

$$\begin{aligned}
 N_t &= (l)4^{l-1} - \frac{1}{3}(4^{l-1} - 1) \\
 &= (l - \frac{1}{3})(4^{l-1}) + \frac{1}{3} \\
 &= \frac{1}{2} n \log_2 n + \frac{2}{3} n + \frac{1}{3} \\
 l(3(4^{l-2})) &= (\frac{1}{2} \log_2 n + 1) \cdot \frac{3}{4} n \\
 &= \frac{3}{8} n \log_2 n - \frac{3}{4} n
 \end{aligned}$$

입으로 식(4)로부터

$$\frac{1}{8} n \log_2 n - \frac{1}{12} n + \frac{1}{3} < N_w < \frac{1}{2} n \log_2 n + \frac{2}{3} n + \frac{1}{3}$$

이다. 따라서 최악의 경우 시간복잡도, N_w 는

$$N_w = O(n \log_2 n) \quad (5)$$

이다.

IV. 결론

본 연구에서는 내장형 영상코딩을 위한 SPIHT(Set Partitioning in Hierarchy Trees) 알고리즘의 실행과정을 SOT 코딩트리 데이터 구조로 새로이 모델화하여 분석하고 처리 시간복잡도(time complexity)를 해석하여 최악의 경우(worst case) 시간복잡도가 $O(n \log_2 n)$ 임을 증명하였다. 부가하여 그림 6과 같은 최악의 경우는 SPIHT 알고리즘은 압축능력을 완전히 상실한다는 사실 또한 확인할 수 있다. SPIHT 알고리즘의 중요성에 비추어 본 시간복잡도 해석 결과는 기존 연구의 비교분석 및 새로운 EZW 계열 알고리즘 개발에 중요한 이론적 배경을 제공할 것으로 기대된다.

접수일자 : 2002. 11. 07 수정완료 : 2003. 2. 11

본 연구는 2001년도 경남대학교 연구년 학술연구 장려금 지원에 의해 수행되었음.

참고문헌

- [1] D. S. Taubman and M. W. Marcellin, JPEG2000: Image Compression Fundamentals, Standards and Practice, Kluwer Academic Publishers, 2002
- [2] J. Y. Tham, S. Ranganath, and A. A. Kassim, "Highly scalable wavelet-based video codec for very low bit-rate environment," IEEE Journal on Selected Area in Communications, vol. 16, pp. 12-27, Jan. 1998.
- [3] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients", IEEE Transactions on Signal Processing, Vol. 41, No. 12, pp. 3445-3462 Dec. 1993.
- [4] Amir Said, William A. Pearlman, A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees, IEEE Transactions on Circuits and Systems for Video Technology, Vol 6, pp. 243-250, 1996.
- [5] David Taubman, "High Performance Scalable Image Compression with EBCOT", IEEE Transaction on Image Processing, Vol. 9, No. 7, July 2000.
- [6] Z. Xiong, K. Ramchandran, and M. Orchard, "Space-frequency Quantization for Wavelet Image Coding", IEEE Trans. on Image Processing, Vol. 6, pp. 677-693, May 1997.
- [7] M. Tsai, J. Villasenor, and F. Chen, "Stack-run Image Coding," IEEE Trans. on Circuits Syst. Video Technol., Vol. 6, pp. 519-521. Oct. 1996.
- [8] Y. W. Chen and W. A. Pearlman, "Three-Dimensional Subband Coding of Video Using the Zero-Tree Method," in symp. on Visual Commun. and Image Processing, SPIE, Vol. 2727, Mar. 1996.
- [9] M. R. Garey and D. S. Johnson, Computer and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Company, 1979.



박영석 (Young-Seak Park)

正會員

1979년 영남대학교 전자공학과

(공학사)

1981년 한양대학교 전자공학과

(공학석사)

1985년 한양대학교 전자공학과 (공학박사)

1990년 ~ 1991년 일본 우정성 통신융합연구소(관서선단연구센터) 초빙과학자

1990년 ~ 1991년 일본 긴끼이동통신센터 객원연구원

2000년 ~ 2001년 미국 North Carolina 주립대학(NCSU) 교환교수

1985년 ~ 현재 경남대학교 정보통신공학부 교수

관심분야: Software Engineering, Web-based Software Design & Development, Pattern Recognition, Image Processing, Computer Network & Network Computing
