

# SPIHT 부호화 모듈의 설계와 구현

장준, 이호석  
호서대학교 컴퓨터 공학과

## A design and implementation of SPIHT encoding module

Jun Chang, Ho Suk Lee  
Department of Computer Engineering, Hoseo University  
changjun@mmlab.hoseo.ac.kr, hslee@office.hoseo.ac.kr

### 요 약

SPIHT(Set Partitioning In Hierarchical Trees) 부호화 알고리즘은 EZW(Embedded Zerotree Wavelet) 부호화 알고리즘의 부호화 방법을 개선하여 압축 효율을 개선한 알고리즘이다. SPIHT 부호화 알고리즘은 웨이블릿 변환된 영상의 계수 값이 동일한 방향을 갖는 대역 사이에서 상관 관계를 갖는다는 점을 이용한다는 점에서 EZW 부호화 알고리즘과 동일하다. 그러나 zerotree의 부호화 부분에서 계수의 중요도에 따라 부분 집합으로 분할해 가는 과정과 분할된 계수들을 부호화하는 과정을 개선하였다. 이 부호화 과정에서의 significant map은 모든 threshold에 대해서 LSP(List of Insignificant Pixels), LIP(List of Insignificant), LIS(List of Insignificant Sets)의 세가지 리스트를 통하여 구하여진다. 그리고, 전체 알고리즘은 초기화, Sorting pass, Refinement pass, 양자화 값 갱신의 네 가지 단계로 구성된다. 본 논문에서는 SPIHT 구현에 필요한 자료구조를 제안하고 SPIHT 부호화 모듈을 구현에 대하여 설명한다.

### 1. 서 론

DCT(Discrete Cosine Transform)을 기반으로 하는 블록 단위의 변환에서는 낮은 비트율에서 블록간의 경계가 현저하게 나타나는 블록경계 현상(blocking artifacts)이 발생한다. 이에 비해 웨이블릿[3] 변환에서는 공간 및 주파수 해상도를 가변적으로 나타낼 수 있어 평탄하게 변하는 부분과 윤곽선과 같이 급격하게 변화하는 부분을 동시에 효과적으로 다룰 수 있으므로 낮은 비트율 압축에서 블록경계 현상을 일으키지 않는다. 웨이블릿 변환을 이용한 부호화 방법인 EZW[2] 영상 부호화 기법은 웨이블릿 변환에 의해 분해된 영상의 자기-유사성을 이용하여 zerotree를 구성하고 영상 정보를 부호화한다.

SPIHT 알고리즘[1]도 위에서 설명한 EZW 알고리즘과 동일하게 zerotree를 이용하여 웨이블릿 계수를 점진적으로 처리한다. 그러나 EZW와의 주요한 차이점은 웨이블릿 계수를 중요도에 따라 부분 집합으로 분할해 가는 과정과 분

할된 계수들을 부호화하는 과정에 있다.

즉, 분할된 계수들의 중요도를 검사하는데 있어, significant map은 모든 threshold에 대해 LSP와 LIS, LIP의 리스트를 통해 구해지고 초기화, Sorting pass, Refinement pass, 그리고 양자화 값 갱신의 단계를 거치게 된다. 초기화 부분에서 최상위의 레벨에 존재하는 계수들의 집합이 초기화되고, Sorting pass에서 최상위 레벨에 존재하는 계수들의 집합의 중요도를 검사하게 된다.

또 최상위 레벨의 계수들의 자식 노드에 대해 직계자손과 자손(직계 자손을 제외한) 노드로 구분하여 중요도를 검사하게 된다. 다음 Refinement pass에서 Sorting pass의 중요도 검사 시 significant로 판정된 계수들에 대해 n번째 MSB(Most Significant Bit)의 값을 전송하게 된다.

본 논문의 구성은 다음과 같다. 다음 2절에서는 SPIHT 알고리즘에 대하여 자세하게 소개한다. 제 3절에서는 SPIHT 알고리즘을 구현하기 위한 자료구조를 자세하게 소개한다. 제 4절에서는 프로그램의 구조와 현재 구현되어 있는 함수에 대

하여 소개한다.

## 2. SPIHT 알고리즘

SPIHT 부호화 알고리즘은 초기화 단계, Sorting pass, Refinement pass, 양자화 값 갱신의 단계로 구성된다.

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

위 식에서 구해진 n 값을 threshold 값으로 초기화하고, threshold 값을 기준으로 양자화 함수  $S_n(\cdot)$  을 다음과 같이 정의한다.

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

SPIHT 부호화 알고리즘의 수행 단계는 다음과 같다.

1. 초기화:

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

1.2 LIP, LSP, LIS 집합 초기화;

2. Sorting pass:

2.1 while (LSP != End of list)

2.1.1 Out =  $S_n(i,j)$ ;

2.1.2 if ( $S_n(i,j) = 1$ ) then (i,j)를 LSP 에 추가,  
 $C_{i,j}$  의 부호 출력;

2.2 while(LIS != End of list)

2.2.1 if (type == A) then

- Out =  $S_n(D(i,j))$ ;

- If ( $S_n(D(i,j)) = 1$ )

■ While( $O(i,j) \neq$  End of list):

◆ Out =  $S_n(k,l)$ ;

◆ if ( $S_n(k,l) = 1$ ) then (k,l)를 LSP 에 추가,  
 $C_{k,l}$  의 부호 출력;

◆ if ( $S_n(k,l) = 0$ ) then (k,l)를 LIP 에 추가;

■ if ( $L(i,j) \neq 0$ ) then (i,j)를 LIS 에 type B 로 추가;

2.2.2 if (type = B) then

- Out =  $S_n(L(i,j))$ ;

- If ( $S_n(L(i,j)) = 1$ ) then

(k,l) ∈  $O(i,j)$ 를 LIS 에 type A 로 추가;

(i,j)를 LIS 에서 삭제;

3. Refinement pass: 지난번 Sorting pass 에서 포함된 것을 제외하고 LSP 의 각 좌표에 대해 n 번째 최상위 비트를 출력;

4. 양자화 값 갱신:  $n = n - 1$ ; go to Step 2;

## 3. 자료 구조 설계

일반적으로 대부분의 영상 정보는 저주파 영역에 집중된다. 그러므로 영상의 주파수가 증가할수록 계수는 감소하게 된다.

이러한 성질을 DWT(Discrete Wavelet Transform)에 이용한다. N-level로 분해된 영상에서 lower level 부분은 더 높은 주파수 서브밴드와 일치한다. 예를 들어, level-1은 가장 높은 주파수 서브밴드이면서, 가장 미세한 영상 정보를 가진다. 그리고 n-th level은 가장 낮은 주파수 서브밴드이면서 가장 중요한 영상 정보들을 가진다. DWT에 의해 변환된 이미지는 이러한 특성과 자기-유사성의 특성을 가지게 된다. 즉  $LH_n, HL_n, HH_n$ 의 화소는  $LH_{n-1}, HL_{n-1}, HH_{n-1}$ 의 화소들의 집합과 상관관계를 가지며 이러한 특성을 이용하여 n-level의 각 화소들은 n-1 level에 대응하는 4개의 화소와 부모-자식간의 관계를 가지는 spatial orientation tree로 구성할 수 있다. 각 화소들은 집합의 개념으로 나눌 수 있는데 다음과 같다.

□  $O(i,j)$ : 노드(i,j)의 모든 자식들의 좌표 집합

□  $D(i,j)$ : 노드(i,j)의 모든 자손들의 좌표집합

□  $H$ : 모든 공간 근원 트리(spatial orientation tree)뿌리들의 좌표 집합

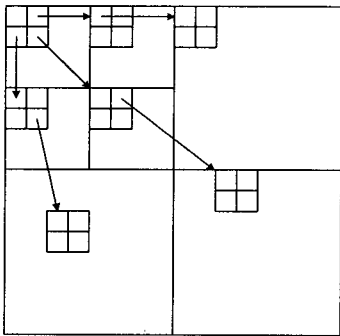
□  $L(i,j) = D(i,j) + O(i,j)$ :

예를 들어, 최상위 피라미드 레벨과 최하위 피라미드 레벨을 제외하고, 다음과 같다.

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

하위집합에서 중요도를 시험하기 위해 3개의 리스트로 구성하게 되는데 LSP와 LIP, LIS로 나눌 수 있다. 3개의 리스트는 연결 리스트로 구성되고, LSP는 '중요하다(significant)'고 판정된 화소의 좌표

를 저장하고, LIP는 초기화시에 가장 낮은



[그림 1] 공간-방향 트리에서의 부모-자식 관계도

단계의 4 개의 화소를 저장하고, 이후 ‘중요하지 않다(insignificant)’고 판정된 화소의 좌표를 저장하게 된다. LIS는 가장 낮은 단계에서 자식을 갖지 않는 좌표를 제외하고, 3 개의 화소를 저장하게 된다. 또 화소의 집합 유형을 나타내기 위해  $D(i,j)$ 에 속하는 유형은 A로 표현하고,  $L(i,j)$ 에 속하는 유형은 B로 표현하게 된다.

#### 4. 프로그램 구조

프로그램은 다음과 같은 구조로 설계되었다.

. LIP.H: LIP 자료구조 선언

```
typedef struct LIP {
    int Coordinate_X;
    int Coordinate_Y;
    struct LIP* next;
} LIP;
```

. LIP.C: LIP 리스트 자료구조의 함수 구현

```
LIP* init_LIP(matrix_2d* m);
int testLIP
(matrix_2d* M, int row, int col, int threshold);
void LIP_addElement(LIP* p, int a, int b);
void LIP_destroyElement(LIP* p, int a, int b);
void LIP_destroyAllElement(LIP* p);
```

. LSP.C: LSP 자료구조의 함수 구현

. LSP.H: LSP 자료구조의 선언과 함수 선언

. LIS.C: LIS 자료구조에 대한 함수 구현

. LIS.H: LIS 자료구조의 선언과 함수 선언

. Matrix2d.H: 입력 이미지를 저장하는 자료구조와 함수 선언

```
typedef struct _matrix_2d {
    int row;
    int col;
    element_type **m;
} matrix_2d;
```

. Matrix2d.C: 자료구조에 대한 함수 구현

. Coder.C:

Sorting pass의 세부절차와

Refinement pass의 세부절차 구현

```
void sigLIS(LIS* lis, matrix_2d* M, LSP* lsp,
LIP* lip, int threshold)
```

```
void sigLIP(LIP* lip, matrix_2d* M, LSP* lsp,
int threshold)
```

#### 5. 결론

본 논문은 8x8의 예제 영상을 통해 SPIHT 구현을 위한 자료구조와 프로그램 구조에 대한 설계를 제시하였다. 설계 방법은 각 화소의 집합을 나타내기 위해 연결 리스트 구조를 사용하였고, 각 자료구조에 대해 초기화, 삽입, 삭제 등의 함수들을 구현하였다. Sorting pass와 Refinement pass는 이들 함수들을 조합하여 구성하였다.

현재 8x8의 예제 영상을 입력 받아 처리하는 부호기를 구현하였으며, 차후 일반적 크기의 영상에 대한 부호기 구현 작업과 복호기 구현 작업이 진행되어야 할 것이다.

#### 6. 참고 문헌

- [1] Amir Said and William A. Pearlman, "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," IEEE Trans. On Circuits and Systems for Video Technology, Vol. 6, June 1996.
- [2] Jerome M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," IEEE Trans. On Signal Processing, Vol. 41, No.12, Dec. pp3445-3462, 1993.
- [3] Raghuvver M. Rao and Ajit S. Bopardikar, Wavelet Transforms—Introduction to Theory and Applications, Addison-Wesley, 1998.