

# 고속 이진 웨이블릿 변환

강의성\*, 이경훈\*\*, 고성제\*\*\*

\*국립순천대학교 컴퓨터교육과, \*\*㈜텔리언, \*\*\*고려대학교 전자공학과

## Fast Binary Wavelet Transform

Eui-Sung Kang\*, Kyung-Hoo Lee\*\*, Sung-Jea Ko\*\*\*

\*Dept. of Computer Education, Suncheon National University,

\*\*Tellion Co.,

\*\*\* Dept. of Electronics Engineering, Korea University

### Abstract

A theory of binary wavelets has been recently proposed by using two-band perfect reconstruction filter banks over binary field. Binary wavelet transform (BWT) of binary images can be used as an alternative to the real-valued wavelet transform of binary images in image processing applications such as compression, edge detection, and recognition. The BWT, however, requires large amount of computations since its operation is accomplished by matrix multiplication. In this paper, a fast BWT algorithm which utilizes filtering operation instead of matrix multiplication is presented. It is shown that the proposed algorithm can significantly reduce the computational complexity of the BWT.

For the decomposition and reconstruction of an  $N \times N$  image, the proposed algorithm requires only  $2LN^2$  multiplications and  $2(L-1)N^2$  additions when the filter length is  $L$ , while the BWT needs  $2N^3$  multiplications and  $2N(N-1)^2$  additions.

### I. 이진 웨이블릿 변환

웨이블릿 변환 (wavelet transform)은 영상압축, 필터링, 에지검출, 시간-주파수 분석 (time-frequency analysis) 등의 분야에서 많은 연구가 이루어져 왔다 [1]. 그 동안 웨이블릿 변환에 대한 연구는 입출력 신호가 실수인 경우에 대해서 주로 이루어져 왔다. 그런데 최근 들어 입출력이 0과 1만으로 이루어진 이진 데이터를 대상으로 하는 이진 웨이블릿 변환 (binary wavelet transform: BWT) 방법이 제안되었다 [2].

BWT는 영상의 분해 (decomposition) 및 복원 (reconstruction) 과정이 GF(2)에 대해서 이루어지는데, 이 과정에서의 모든 연산은 AND, OR, XOR 등과 같은 논리 연산에 의해서 가능하다. 따라서 변환과정이 단순할 뿐만 아니라, 특히 하드웨어로 구현할 경우에는 속도면에서 큰 이점을 얻을 수 있다.

또한, 팩시밀리나 지문 등의 이진 데이터를 BWT에 의해서 다해상도로 분해한 결과는 동일한 데이터를 실수 (float) 웨이블릿 변환한 후, 이를 thresholding한 결과와 아주 유사한 결과를 얻을 수 있다. 따라서, 이진 영상에 대한 에지 검출(edge detection), 특징소 추출

(feature extraction), 전처리 (preprocessing) 등의 분야에 효과적으로 적용될 수 있다.

그런데, BWT에서는 영상의 분해는 저주파 필터와 고주파 필터에 의해서 구성된 변환 행렬 (transform matrix)  $T$  와 이에 대한 전치행렬 (transpose matrix)  $T^T$  를 원영상  $F$  의 좌우에 곱함으로써 수행되고, 복원은 변환 행렬에 대한 역행렬  $T^{-1}$  과 이에 대한 전치행렬  $(T^{-1})^T$  을 변환 영상의 좌우에 곱함으로써 수행된다.

이와 같은 BWT의 분해 및 복원 과정 중, 저주파 필터 및 고주파 필터에 의한 2-band 필터뱅크 구조로 대체할 수 있지만, 복원 과정에서는 실수 웨이블릿 변환과는 달리 복원을 위한 이진 필터가 알려져 있지 않으므로, 역행렬에 의한 행렬 곱셈을 이용해야 한다. 일반적으로 행렬 곱셈은 행렬의 차원 (dimension)이 커질수록 계산량이 매우 증가하게 되는데, 영상처리에서와 같이 입력 데이터의 차원이 큰 경우에는 복원 과정이 매우 비효율적이다. 따라서 BWT를 다양한 응용 분야에 적용하기 위해서는 행렬 곱셈을 대체할 수 있는 고속 알고리즘이 필수적으로 요구된다.

본 논문에서는 이진 웨이블릿 변환을 위한 변환 행렬에 대한 역행렬  $T^{-1}$  이 희소 행렬 (sparse matrix)이며, 역행렬의 각 행이 일정한 패턴을 유지한다는 사실을 이용하여 행렬 곱셈에 의한 BWT의 복원 과정을 컨볼루션 (convolution) 형태로 대체함으로써 계산량을 줄일 수 있는 고속 BWT 알고리즘을 제안한다.

### II. 이진 웨이블릿 변환

$N \times N$  행렬  $A$  가 있을 때,  $A$  의  $i$  번째 행이 첫번째 행을  $(i-1)$  번 쉬프트 (shift)한 것으로 구성될 때, 이러한 행렬을 circulant 행렬이라고 하는데, 쉬프트 되는 양이 1인 행렬을 one-circulant 행렬, 2인 행렬을 two-circulant 행렬이라고 한다. 예를 들어 행렬  $A$  가 two-circulant 행렬이라면 다음과 같은 형태를 갖는다.

$$A = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_i & \cdots & a_{N-2} & a_{N-1} \\ a_{N-2} & a_{N-1} & a_0 & a_1 & a_2 & \cdots & a_{N-4} & a_{N-3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{N-(2i-2)} & a_{N-(2i-3)} & \cdots & \cdots & \cdots & \cdots & a_{N-2} & a_{N-(2i-1)} \\ a_{N-2i} & a_{N-(2i-1)} & \cdots & \cdots & \cdots & \cdots & a_{N-(2i+2)} & a_{N-(2i+1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_2 & a_3 & a_4 & \cdots & \cdots & \cdots & a_0 & a_1 \end{bmatrix} \quad (1)$$

이진 웨이블릿 변환을 위한 저주파 필터  $l=[l_0, l_1]$ , 고주파 필터  $h=[h_0, h_1]$  라고 하고, 이 두 필터에 의해서 구성되는  $N/2 \times N$  two-circulant 행렬을 각각  $L_2, H_2$  라고 하자.

$$L_2 = \begin{bmatrix} l_0 & l_1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & l_0 & l_1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & l_0 & l_1 \end{bmatrix} \quad (2)$$

$$H_2 = \begin{bmatrix} h_0 & h_1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & h_0 & h_1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & h_0 & h_1 \end{bmatrix} \quad (3)$$

이진 웨이블릿 변환을 위한  $N \times N$  변환 행렬  $T$  는 식 (4)와 같은 형태를 갖는다.

$$T = \begin{bmatrix} L_2 \\ H_2 \end{bmatrix} = \begin{bmatrix} l_0 & l_1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & l_0 & l_1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & l_0 & l_1 \\ h_0 & h_1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & h_0 & h_1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & h_0 & h_1 \end{bmatrix} \quad (4)$$

위와 같은 변환 행렬  $T$  와 이진 입력 영상  $F=[f_{jk}]_{j,k=0,\dots,N-1}$  를 곱하면 다음과 같은 식을 얻을 수 있는데, 이 식의 결과는 수직방향으로 이진 웨이블릿 변환한 영상을 나타냄을 알 수 있다.

$$T \cdot F = \begin{bmatrix} l_0 & l_1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & l_0 & l_1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & l_0 & l_1 \\ h_0 & h_1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & h_0 & h_1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & h_0 & h_1 \end{bmatrix} \begin{bmatrix} f_{00} & \cdots & f_{0(N-1)} \\ f_{10} & \cdots & f_{1(N-1)} \\ f_{20} & \cdots & f_{2(N-1)} \\ f_{30} & \cdots & f_{3(N-1)} \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ f_{(N-2)0} & \cdots & f_{(N-2)(N-1)} \\ f_{(N-1)0} & \cdots & f_{(N-1)(N-1)} \end{bmatrix} \quad (5)$$

즉, 변환 행렬  $T$  와 입력 영상  $F$  의 곱  $T \cdot F$  는 아래 그림과 같은 필터뱅크 구조를 이용하여 입력 영상을 수직 방향으로 필터링한 후, 다운샘플링한 결과와 일치한다.

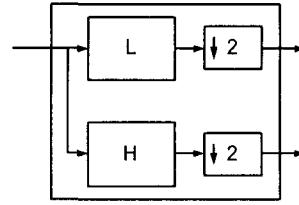


그림 1. 필터뱅크 구조

그리고, 입력 영상을 수평 방향으로 변환하기 위해서는 입력 영상  $F$  의 오른쪽에  $T^T$  를 곱하면 된다. 결국, 식 (6)과 같이  $T$  와  $T^T$  를 입력 영상의 양쪽에 곱하면 수직, 수평 방향으로 이진 웨이블릿 변환된 영상을 얻을 수 있다.

$$Y = T \cdot F \cdot T^T \quad (6)$$

위와 같은 이진 웨이블릿 변환을 위한 two-circulant 변환 행렬  $T$  는 역행렬을 갖는다고 알려져 있다 [2]. 이진 웨이블릿 변환된 영상을 복원하기 위해서는 분해 과정과 동일한 방법으로 식 (7)과 같이 역행렬을 변환 영상의 양쪽에 곱해줌으로써 원영상을 복원할 수 있다.

$$R = T^{-1} \cdot Y \cdot (T^{-1})^T = T^{-1} \cdot T \cdot F \cdot T^T \cdot (T^{-1})^T = F \quad (7)$$

### III. 제안한 고속 이진 웨이블릿 변환

이진 웨이블릿에 의한 영상의 분해 과정은 행렬 곱셈에 의한 방법 또는 필터뱅크 구조로 이용한 필터링 과정에 의해서 수행될 수 있으나, 분해 과정과는 달리 복원 과정을 위한 이진 웨이블릿 변환 필터가 제안되어 있지 않다. 따라서, 복원 과정에서는 필수적으로 역행렬에 의한 행렬 곱셈 과정을 거쳐야 하는데, 이 과정에서 많은 계산량이 요구된다. 예를 들어,  $N \times N$  차원을 갖는 행렬의 곱셈  $C = A \cdot B$  를 고려할 때,  $c_{ij} = \sum_k a_{ik} b_{kj}$  이

므로, 하나의 원소(픽셀)에 대한 결과를 얻는 데에는  $N$  번의 곱셈과  $N-1$  번의 덧셈이 요구된다. 이에 비해서 필터링을 이용한 경우에는 동일한 결과를 얻기 위해서  $L$  번의 곱셈과  $L-1$  번의 덧셈이 요구된다. 여기서,  $L$  은 필터 계수의 길이이다. 일반적으로 필터의 길이에 비해서 입력 영상의 크기가 아주 크기 때문에 행렬에 의한 복원 방법은 필터링에 의한 방법에 비해서 매우 비효율적이다. 또한, 입력 영상이 커짐에 따라서 행렬에 의한 복원 과정에서 소요되는 계산량이 기하급수적으로 증가하게 된다.

본 논문에서는 행렬 곱셈에 의한 이진 웨이블릿 복원 과정에서의 비효율성을 극복하기 위해서 분해 과정에서의 마찬가지로 필터링 연산에 의한 고속 이진 웨이블릿 복원 방법을 제안한다. 제안한 방법에서는 이진 웨이블릿 복원을 위한 역변환 행렬 (inverse transform matrix)  $T^{-1}$  이 희소 행렬이 된다는 사실을 이용한다.

입력 영상의 크기가  $8 \times 8$  이고, 이진 웨이블릿 변환을 위한 저주파 필터 및 고주파 필터를 각각  $l=[1,0]$ ,  $h=[1,1]$  라 하면 변환 행렬에 대한 역행렬  $T^{-1}$  은 다음

식과 같다.

$$\mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$\mathbf{T}^{-1}$  은 식 (9)와 같이  $\mathbf{T}_L^{-1}$  과  $\mathbf{T}_R^{-1}$  으로 분리하여 쓸 수 있다.

$$\mathbf{T}^{-1} = \mathbf{T}_L^{-1} + \mathbf{T}_R^{-1} \quad (9)$$

여기서,  $\mathbf{T}_L^{-1}$ ,  $\mathbf{T}_R^{-1}$  은 각각 다음과 같다.

$$\mathbf{T}_L^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{T}_R^{-1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

식 (6)에 의해서 수직, 수평 방향으로 변환된 영상  $\mathbf{Y}$  를 주파수 대역별로 식 (11)과 같이 나타내기로 하자.

$$\mathbf{Y} = \begin{bmatrix} \mathbf{LL} & \mathbf{LH} \\ \mathbf{HL} & \mathbf{HH} \end{bmatrix} \quad (11)$$

식 (7)에 의한 영상 복원을 위한 행렬식  $\mathbf{R} = \mathbf{T}^{-1} \cdot \mathbf{Y} \cdot (\mathbf{T}^{-1})^T$  에서  $\mathbf{T}^{-1} \cdot \mathbf{Y}$  은 식 (9)에 의해서  $\mathbf{T}_L^{-1} \cdot \mathbf{Y} + \mathbf{T}_R^{-1} \cdot \mathbf{Y}$  으로 나타낼 수 있다. 즉,

$$\mathbf{T}^{-1} \cdot \mathbf{Y} = \mathbf{T}_L^{-1} \cdot \begin{bmatrix} \mathbf{LL} & \mathbf{LH} \\ \mathbf{HL} & \mathbf{HH} \end{bmatrix} + \mathbf{T}_R^{-1} \cdot \begin{bmatrix} \mathbf{LL} & \mathbf{LH} \\ \mathbf{HL} & \mathbf{HH} \end{bmatrix} \quad (12)$$

이때,  $\mathbf{T}_L^{-1}$  열의 오른쪽 반 (right-half)과,  $\mathbf{T}_R^{-1}$  열의 왼쪽 반 (left-half)이 모두 영이므로, 식 (12)는 다음과 같이 쓸 수 있다.

$$\mathbf{T}^{-1} \cdot \mathbf{Y} = \mathbf{T}_L^{-1} \cdot \begin{bmatrix} \mathbf{LL} & \mathbf{LH} \\ 0 & 0 \end{bmatrix} + \mathbf{T}_R^{-1} \cdot \begin{bmatrix} 0 & 0 \\ \mathbf{HL} & \mathbf{HH} \end{bmatrix} \quad (13)$$

식 (13)은 [1 0] 을 필터 계수로 하는 필터링을  $\mathbf{LL}$  과  $\mathbf{LH}$  대역에 대해서 수행하고, [1 1] 을 필터 계수로 해서  $\mathbf{HL}$  과  $\mathbf{HH}$  대역에 대해서 필터링을 수행함으로써 행렬 곱셈을 대체할 수 있다.  $\mathbf{R} = \mathbf{T}^{-1} \cdot \mathbf{Y} \cdot (\mathbf{T}^{-1})^T$  에서의  $\mathbf{Y} \cdot (\mathbf{T}^{-1})^T$  계산 또한 동일한 방법에 의해서 행렬 연산 을 필터링 연산으로 대체할 수 있다.

행렬에 의한 복원 과정에서 또다른 문제점은 입력되는 영상의 크기 (dimension)가 달라지면, 영상의 크기에 따라서 변환 행렬  $\mathbf{T}$  에 대한 역행렬을 새로이 구해야 된다는 단점이 있다. 그런데, 그 역행렬은 차원이 달라질 뿐 식 (10)에서 같이 일정한 패턴이 규칙적으로 반복된다. 따라서 행렬 곱셈에 의한 방법과는 달리 영상의 크기에 따라서 역행렬 계산과 같은 부가적인 계산량이 요구되지 않는다는 장점이 있다.

#### IV. 실험 및 결과

본 논문에서는 256×256 크기의 이진 영상을 입력 영상으로 하여 BWT에서 제안된 세 가지의 필터 [2]에 대해서 실험하였다. 그림 2는 표 1에 제시된 이진 웨이블릿 필터를 가지고 타이어 (tire) 영상을 3번 대역 분할된 결과를 보여 주고 있다. 그림 2(a)는 원영상이고, (b)~(d)는 필터 계수의 길이에 따른 대역 분해 결과이다. 그림에서 볼 수 있는 바와 같이 이진 웨이블릿 변환에 의한 결과가 일반적인 실수 웨이블릿 변환 결과를 thresholding한 결과와 매우 유사한 특징을 보임을 알 수 있다.

표 1. 이진 웨이블릿 필터

	$L=2$	$L=4$	$L=8$
Lowpass filter	[1 0]	[1 1 1 0]	[1 1 1 0 1 0 1 0]
Highpass filter	[1 1]	[1 1 0 0]	[1 1 1 1 1 0 0]

앞서서 언급한 바와 같이 행렬 연산에 의한 이진 웨이블릿 변환을 필터링 연산으로 변환할 수 있는 이유는 변환 영상을 복원하는데 쓰이는 변환 행렬에 대한 역행렬이 규칙적인 패턴을 갖는 희소 행렬을 이루기 때문이다. 그런데, 표 1에 제시된 이진 웨이블릿 필터 중에서 필터의 길이가 8인 경우에는 변환 행렬에 대한 역행렬이 규칙적인 패턴을 갖기는 하지만 희소행렬을 이루지 않으므로 인해서 제안된 고속 알고리즘을 적용할 수 없다. 실수 웨이블릿 변환의 경우에는 고주파 대역 내의 웨이블릿 계수들이 저주파 대역의 계수에 비해서 그 크기가 상대적으로 작은 값을 갖는 경향이 있으나, 이진 웨이블릿 변환의 경우에는 연산이 GF(2)에서의 modulo-2 연산에 의존하기 때문에 고주파 대역 내의 계수들 또한 저주파 대역과 마찬가지로 0 또는 1의 값을 갖는다. 대역 분해에 사용되는 필터 계수의 길이가 길수록, 필터링 결과에 영향을 미치는 픽셀들의 증가하게 되므로, 고주파 대역 내의 계수들이 0이 아닌 값을 가질 가능성이 커지게 된다. 필터의 길이가 8인 그림 2(d)을 살펴보면 대부분의 고주파 대역에도 값이 1인 계수들이 상당히 존재함을 확인할 수 있다. 이러한 사실은 이진 영상에 대한 에지 검출, 특징소 추출, 압축, 등의 부호화에 바람직하지 않은 영향을 미칠 수 있다. 필터 계수의 길이가 2와 4인 경우를 살펴 보면, 원영상 내의 고주파 성분에 해당하는 에지 성분을 적절히 잘 나타내고 있음을 볼 수 있다.

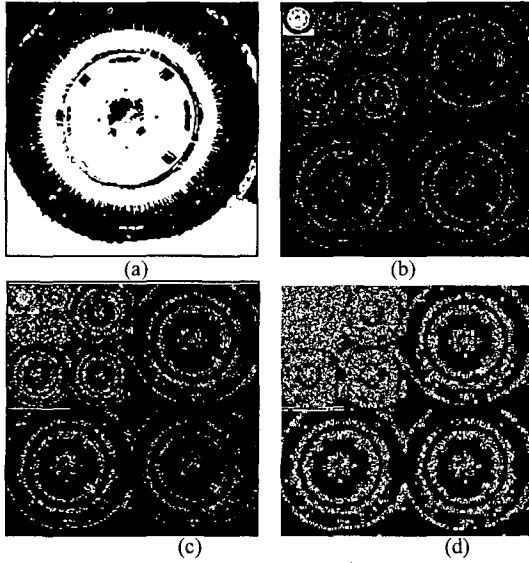


그림 2. 필터 계수에 따른 이진 웨이블릿 변환 영상: (a) 테스트 영상, (b) 필터의 길이가 2인 경우 ( $L=2$ ), (c) 필터의 길이가 4인 경우 ( $L=4$ ), 필터의 길이가 8인 경우 ( $L=8$ )

영상의 크기가  $N \times N$  이고, 필터 계수의 길이가  $L$  일 때, 기존의 행렬 연산에 의해서 이진 웨이블릿 복원시 요구되는 계산량은 식 (7)에 나타난 행렬식을 계산하는데 소요되는 곱셈과 덧셈의 수이다. 그리고, 제안한 방법에 대한 계산량은 식 (13)에 의해서  $T^{-1} \cdot Y$ 를 계산하는데 요구되는 곱셈과 덧셈의 수와, 같은 방법을 이용하여  $Y \cdot (T^{-1})^T$ 를 계산하는데 요구되는 곱셈과 덧셈의 수의 합이다.

표 2는 웨이블릿 복원시에 두 가지 방법에 대한 계산량을 비교한 것이다. 일반적으로 필터의 길이  $L$ 은 입력 영상의 가로 또는 세로 길이에 비해서 훨씬 작으므로, 제안된 방법은 원영상을 복원하는데 요구되는 계산량을 크게 줄일 수 있음을 알 수 있다. 분해 과정에서의 계산량은 기존의 BWT에서도 필터 뱅크 구조를 이용할 수 있으므로 동일한 계산량을 갖는다.

표 2. 이진 웨이블릿 변환 (복원시) 과정의 계산량 비교

	Multiplications	Additions
BWT	$2N^3$	$2(N-1)N^2$
Proposed method	$2LN^2$	$2(L-1)N^2$

## V. 결론

본 논문에서는 이진 웨이블릿 변환에 대한 고속 알고리즘을 제안하였다. 기존의 이진 웨이블릿 변환 방법이 행렬 연산에 의해서 수행됨으로 인해서 계산량이 많은 반면, 제안한 방법에서는 변환 행렬에 대한 역행렬이 규칙적인 패턴이 반복되는 희소 행렬이 된다는 사실을 이용하여 행렬에 의한 영상의 복원 대신에 필터링 연산을 이용하여 변환 영상으로부터 원영상을 복원하기 때문에,

계산량을 크게 줄일 수 있다. 이러한 이진 웨이블릿 변환은 입출력 뿐만 아니라, 처리 과정에서의 데이터들이 모두 이진값을 가지므로, 하드웨어로 구현할 경우 논리 게이트를 이용하여 구현 가능하여 다양한 응용분야에서 유용하게 적용될 수 있다

## 참고문헌

- [1] R. M. Rao and A. S. Bopardikar, "Wavelet Transforms - Introduction to theory and applications," Addison Wesley, 1998.
- [2] M. D. Swanson and A. H. Tewfik, "A Binary wavelet decomposition of binary images," *IEEE Trans. Image Processing*, vol. 5, pp. 1637-1650, Dec. 1996.
- [3] S. J. Leon, "Linear algebra with applications," Prentice-Hall, 1994.
- [4] S. Nakamura, "Applied numerical method in C," Prentice-Hall, 1993.